UNIVERSITÉ DE MONTRÉAL

ICE INTERFACE EVOLUTION MODELLING ALGORITHMS FOR AIRCRAFT ICING

SIMON BOURGAULT-CÔTÉ
DÉPARTEMENT DE GÉNIE MÉCANIQUE
ÉCOLE POLYTECHNIQUE DE MONTRÉAL

THÈSE PRÉSENTÉE EN VUE DE L'OBTENTION
DU DIPLÔME DE PHILOSOPHIÆ DOCTOR
(GÉNIE MÉCANIQUE)
FÉVRIER 2019

UNIVERSITÉ DE MONTRÉAL

ÉCOLE POLYTECHNIQUE DE MONTRÉAL

Cette thèse intitulée :

ICE INTERFACE EVOLUTION MODELLING ALGORITHMS FOR AIRCRAFT ICING

présentée par : BOURGAULT-CÔTÉ Simon
en vue de l'obtention du diplôme de : Philosophiæ Doctor
a été dûment acceptée par le jury d'examen constitué de :

M. DUFOUR Steven, Ph. D., président
M. LAURENDEAU Éric, Ph. D., membre et directeur de recherche
M. PIPERNI Pat, Ph. D., membre
M. RADENAC Emmanuel, Doctorat, membre externe

DEDICATION

To Ariane,
who is always there
when I need her most...

ACKNOWLEDGEMENTS

The research work described in this thesis is part of a project named "Aerodynamic Flow Modelling using RANS of Icing and Transitional Flows on 3D High-Lift Aircraft Configurations" funded by a Bombardier/NSERC Collaborative Research and Development (CRD) grant. The author was funded by a "NSERC Postgraduate Scholarships - Doctoral Program" from Canada. The contribution of the foundation Friends of Polytechnique through a grant "Bourse Prestige, États-Unis" won by the author was also decisive in the realization of the project by funding an internship at the Massachusetts Institute of Technology.

To break the ice, I would like to thank my supervisor, Prof. Éric Laurendeau, who provided me with just enough guidance to align me in a good direction while letting me find the correct path. I must say that finding any path in the aircraft icing field is hazardous and slippery at best, but he trusted me and I am grateful for that. He provided me with multiple opportunities, such as internships at Bombardier and MIT, and allowed me to go to many conferences. He was always available when I needed help for the project and his interest on the topic motivated me in my work. For all this and more, thank you very much Éric.

I would also like to thank the people from the Advanced Aerodynamics department at Bombardier involved in the project, in particular Hong Yang and Guy Fortin. The discussions we had during my internships at Bombardier on programming and icing were very helpful for my research, but even more for my own development as a programmer and engineer.

I also need to thank Bob Haimes who accepted me in his team at MIT for the crazy project of integrating geometry within icing. He helped me a lot by always asking questions for which I had no answers. Within his team, I am very grateful for the help and friendship I received from Julia Docampo-Sànchez.

I want to also express a special thanks to some members of the lab at Polytechnique Montreal: i) Kazem for his advice on icing and mesh generation and his general support, ii) Pierre for his discussions about thermodynamic and icing in general, iii) Matthieu for the coffee breaks and the other talks about pretty much anything and iv) Réda for thoroughly reading and commenting on this thesis.

I definitely thank my parents, Suzanne and Léon, who supported me since the beginning. Without their guidance, I would not have reached that height.

I have a final and special thanks for Ariane who waited for me to complete my studies without complaint. Thanks also for the ice shape drawings. Mostly, thanks for being you.

RÉSUMÉ

Le givrage des aéronefs est un phénomène qui nuit à la sécurité des vols depuis le début de l'aviation et qui n'est toujours pas bien compris. Ainsi, des réglementations ont été définies pour identifier les conditions de givrage contre lesquelles les avions doivent être protégés, mais les conditions les plus critiques n'ont été ajoutées que récemment dans la réglementation après des accidents majeurs. Ces conditions critiques incluent l'accumulation de verglas sur les ailes en flèche et l'agglomération de cristaux de glace, ce qui est particulièrement problématique pour les moteurs. Ces domaines font l'objet de recherches expérimentales, mais le développement de modèles et d'outils appropriés pour simuler ces conditions numériquement est également important. Ces simulations comprennent plusieurs modules exécutés de façon séquentielle pour modéliser l'écoulement aérodynamique, les phases dispersées (gouttelettes et cristaux), les échanges thermodynamiques de surface et l'évolution de la géométrie. Ce dernier point est moins étudié dans la littérature que les autres et dépend i) d'une représentation géométrique et ii) d'une méthode permettant de faire évoluer le front de glace.

L'objectif principal de ce travail est de fournir un ensemble approprié de méthodes pour simuler l'accrétion de glace avec un logiciel multicouche automatisé. Dans ce contexte, le module d'évolution de la géométrie est identifié comme étant la partie faible devant être améliorée. Les développements sont réalisés en deux dimensions dans le logiciel maison NSCODE-ICE, mais les méthodes étudiées sont considérées pour une future extension en trois dimensions. De ce fait, la robustesse des méthodes est de la plus haute importance, en particulier lors de la phase de transition entre deux couches de glace. La méthode d'évolution de la géométrie développée devrait être capable de gérer la fusion des fronts de glace tout en conservant la masse de glace. L'accumulation de glace étant par nature chaotique, il serait également intéressant d'étudier l'impact d'un modèle stochastique sur les résultats.

Premièrement, les différents composants d'un logiciel d'accrétion de glace sont détaillés pour établir la base du logiciel de givrage multicouche utilisé dans cette étude. L'écoulement aérodynamique est simulé avec un solveur RANS structuré et les trajectoires des gouttelettes sont modélisées avec une formulation eulérienne sur la même grille. Les échanges thermodynamiques en surface et les eaux de ruissellement sont modélisées avec un modèle de Messinger itératif. Une évolution algébrique standard de la géométrie fait partie de NSCODE-ICE et est utilisée comme point de comparaison pour les autres méthodes de cette étude. Différentes stratégies sont appliquées pour accroître la robustesse du processus de givrage multicouche : i) les B-splines sont introduites en tant que support géométrique, ii) une méthode d'approxi-

mation originale est développée pour transférer de manière adéquate les caractéristiques complexes des formes de glace dans les B-splines, iii) une discrétisation de surface basée sur la courbure est mise en œuvre pour discrétiser correctement les points convexes et concaves autour des formes de glace et iv) une méthode de génération de maillage hyperbolique est ajoutée au programme pour générer des grilles de meilleure qualité.

Deuxièmement, une méthode d'accrétion de glace par niveaux (*level-set*) est introduite pour faire évoluer la géométrie de la glace, sa formulation implicite permettant de gérer automatiquement la fusion de fronts. Une formulation formelle de la vitesse du front est proposée pour permettre la conservation de la masse de glace et une méthode de suivi explicite est mise en œuvre pour augmenter la robustesse de la phase d'extraction de la géométrie. Ce suivi explicite est également utilisé pour contrôler activement l'aire balayée globale afin de garantir la conservation de la masse de glace au niveau global. La méthode est étudiée sur des cas canoniques et expérimentaux et comparée à la méthode algébrique. Les résultats montrent que le contrôle actif de masse de glace fonctionne avec une marge d'erreur de 0,05%.

Troisièmement, une méthode originale permettant de faire évoluer la géométrie de façon explicite est développée sur la base des équations de génération de maillage hyperbolique, qui sont reconnues dans ce travail pour produire des grilles contenant des caractéristiques semblables à la glace. Un nouveau terme source est ajouté pour utiliser la solution de l'équation *Eikonal* afin d'éviter le chevauchement d'un front de glace avec la géométrie. Les différents mécanismes de lissage inclus dans la méthode en font une approche très robuste, même dans les régions concaves, ce qui est démontré pour des cas canoniques. De plus, la nature de maillage de la méthode permet de contrôler activement la conservation de la masse de glace à l'échelle globale, mais également à l'échelle locale. Les résultats sont présentés sur des cas expérimentaux avec des erreurs numériques globales de masse de glace inférieures à 0,01%.

Développée autour d'un concept différent, une approche non déterministe est ensuite présentée pour modéliser l'accumulation de glace avec de la stochasticité. L'approche est basée sur un modèle connu de la littérature, mais ses composants centraux sont modifiés pour utiliser les points forts de NSCODE-ICE, tels que le solveur de gouttelettes eulérien pour la probabilité d'impact ou la capacité d'utiliser des maillages superposés pour les interpolations entre les maillages. Une grille cartésienne bâtie par avance de front originale est développée pour augmenter de manière significative l'efficacité de la méthode, qui construit la glace pixel par pixel. Couplée aux B-splines, cette grille bâtie par avance de front permet de réaliser des simulations multicouches avec la méthode stochastique, ce qui n'est pas fait dans la littérature. L'approche n'est pas encore prête pour les cas de verglas et les résultats ne sont donc présentés que sur des cas expérimentaux de givre. Les résultats moyennés montrent que la

méthode peut converger vers une forme de glace lors de l'ajout de plusieurs couches, même si elle est stochastique.

Pour compléter l'étude, une comparaison est effectuée entre les méthodes d'évolution de la géométrie déterministes incluses dans cette étude : les méthodes i) algébrique, ii) par niveaux (*level-set*) et iii) hyperbolique. Des métriques quantitatives et des critères qualitatifs sont présentés et sont utilisés pour étudier les caractéristiques des résultats obtenus sur deux cas expérimentaux. Les métriques quantitatives ne permettent pas d'identifier de tendances, sauf que les trois méthodes peuvent être globalement conservatrices, mais les critères qualitatifs montrent que les trois méthodes convergent vers la même solution avec des simulations composées de suffisamment de couches de glace. Les approches sont ensuite comparées sur la base des développements numériques et de la robustesse en vue de l'extension en trois dimensions et la méthode d'accrétion de glace hyperbolique est identifiée comme ayant le potentiel le plus important à cette fin.

En conclusion, différentes méthodes ont été développées pour modéliser le processus d'évolution du front de glace, chacune avec ses forces et ses faiblesses. La méthode identifiée comme étant la plus robuste est la méthode d'accrétion de glace hyperbolique qui est un développement original introduit dans cette thèse. Les résultats montrent que pour chaque méthode, l'utilisation d'une représentation géométrique continue telle que les B-splines augmente la robustesse et permet d'effectuer des simulations multicouches de façon automatique. Les B-splines pourraient être utilisées de manière plus intégrée dans la méthode hyperbolique pour gérer la fusion des fronts de glace, ce qui reste l'une des limites de la méthode en raison de sa formulation explicite.

ABSTRACT

Aircraft icing is a phenomenon that has hindered flight safety since the beginning of aviation and that is still not well understood. Thus, regulations have been set to identify icing conditions that the aircraft must be protected against, but the most critical conditions were only added recently in the regulations after major accidents happened. These critical conditions include scallops ice accretion on swept wings and ice crystal agglomerations, which is particularly problematic for engines. These fields are being researched experimentally, but developing appropriate models and tools to simulate these conditions numerically is also of importance. Such simulations include multiple modules run sequentially to model the aerodynamic flow, the dispersed phases (droplets and crystals), the surface thermodynamic exchanges and the geometry evolution. This last point is less studied in the literature than the others and depends on i) a geometry representation and ii) a method to evolve the ice front.

The main objective of this work is to provide an appropriate set of methods to simulate ice accretion with an automated multi-layer software. In this context, the geometry evolution module is identified as the weak part which needs to be improved. The developments are done in two dimensions within the in-house ice accretion tool NSCODE-ICE, but the methods studied are considered for a future three-dimensional extension. As such, the robustness of the methods is of utmost importance, in particular within the transition phase between two ice layers. The developed geometry evolution method should be able to handle merging ice fronts while still conserving the ice mass. Ice accretion being chaotic by nature, to study the impact of a stochastic model on the results would also be of interest.

First, the different components of an ice accretion software are detailed to establish the basis of the multi-layer icing software used in this study. The aerodynamic flow is simulated with a structured RANS solver and the droplets trajectories are modelled with an Eulerian formulation on the same grid. The thermodynamic exchanges on the surface and the runback water are modelled with an iterative Messinger model. A standard algebraic geometry evolution is part of NSCODE-ICE and is used as a comparative point for the other methods in this study. Different strategies are applied to increase the robustness of the multi-layer icing process : i) B-splines are introduced as geometry support, ii) an original approximation method is developed to transfer adequately the complex features of ice shapes into B-splines, iii) a curvature-based surface discretization is implemented to correctly discretize the convex and concave points around the ice shapes and iv) a hyperbolic mesh generation method is

added to the software to generate grids with higher quality.

Second, a level-set ice accretion method is introduced to evolve the ice geometry as its implicit formulation allows to automatically handle merging fronts. A formal front velocity formulation is proposed to allow for ice mass conservation and an explicit tracking method is implemented to increase the robustness of the geometry extraction phase. This explicit tracking is also used to actively control the global swept area to ensure the ice mass is conserved globally. The method is studied on canonical and experimental cases and compared to the algebraic method. The results show that the ice mass active control is working within an error margin of 0.05%.

Third, an original method to evolve a geometry explicitly is developed based on the hyperbolic mesh generation equations, which are recognized in this work to produce grids which contain ice-like features. A new source term is added to make use of the Eikonal equation solution to avoid an overlap of an ice front with the geometry. The different smoothing mechanisms included in the method make for a very robust approach even in concave regions, which is shown on canonical cases. Furthermore, the grid nature of the method allows to actively control the ice mass conservation in a global scale, but also in a local scale. Results are presented on experimental cases with global numerical ice mass errors below 0.01%.

Developed around a different concept, a non-deterministic approach is then presented to model the ice accretion with stochasticity. The approach is based on a known model from the literature, but its core components are modified to use the strengths of NSCODE-ICE, such as the Eulerian droplet solver for the impingement probability or the overset grid capability for interpolations between grids. An original advancing front Cartesian grid is developed to increase significantly the efficiency of the method, which builds ice in a pixel by pixel manner. Coupled with the B-splines, this advancing front grid allows to perform multi-layer simulations with the stochastic method, which is not done in the literature. The approach is not yet ready for glaze ice cases and results are thus only presented on rime ice experimental cases. The averaged results show that the method can converge toward an ice shape when adding multiple layers even though it is stochastic.

To complete the study, a comparison is performed between the deterministic geometry evolution methods included in this study : i) the algebraic, ii) level-set and iii) hyperbolic methods. Quantitative metrics and qualitative criteria are presented and are used to study the characteristics of the results obtained on two experimental cases. The quantitative metrics do not allow to identify any trends, except that all three methods can be globally mass conservative, but the qualitative criteria show that the three methods converge toward the same solution with enough ice layers in simulations. The approaches are then compared on the basis of

numerical implementation and robustness in view of a three-dimensional extension and the hyperbolic ice accretion method is identified as having a greater potential to that end.

To conclude, different methods have been developed to model the ice front evolution process, each with different strengths and weaknesses. The one which is found to be the most robust is the hyperbolic ice accretion, an original development introduced in this thesis. For each method, the results show that using a continuous geometry , such as B-splines, increases the robustness and allows to perform multi-layer simulations automatically. The B-splines could be used in a more integrated manner within the hyperbolic method to handle merging ice fronts, which is still one of the limitations of the method due to its explicit formulation.

TABLE OF CONTENTS

DEDICATION III

ACKNOWLEDGEMENTS IV

RÉSUMÉ V

ABSTRACT VIII

TABLE OF CONTENTS XI

LIST OF TABLES XV

LIST OF FIGURES XVI

LIST OF SYMBOLS AND ABBREVIATIONS XXII

LIST OF APPENDICES XXIII

CHAPTER 1 INTRODUCTION 1
1.1 Aircraft icing 1
1.1.1 Regulations and certification 3
1.2 Ice morphologies 5
1.2.1 Airfoil ice 5
1.2.2 Surface roughness and micro-scale 6
1.2.3 Ice crystal agglomeration 7
1.2.4 Swept-wing ice 7
1.3 Icing numerical modelling 9
1.3.1 Geometry Representation 11
1.3.2 Ice evolution method 12
1.4 Objectives of present study 13
1.5 Thesis outline 14

CHAPTER 2 MULTI-LAYER ICING SIMULATION 15
2.1 Numerical simulation process 15
2.1.1 Simplifying hypothesis 17
2.2 Continuous phase modelling 18

2.2.1 Reynolds Averaged Navier-Stokes (RANS) solver 18
2.2.2 Convective heat transfer coefficient 19
2.2.3 Surface roughness 20
2.3 Dispersed phase modelling 20
2.3.1 Eulerian droplet solver 22
2.4 Thermodynamic surface exchanges 23
2.4.1 Iterative Messinger model 25
2.5 Geometry evolution 26
2.5.1 Iterative algebraic method 28
2.6 Volume mesh discretization 29
2.6.1 Elliptic mesh generation 29
2.6.2 Overset grid method 30
2.7 Strategies for robustness 31
2.7.1 Continuous geometry support 31
2.7.2 B-Spline approximation method 32
2.7.3 Curvature-based surface discretization 37
2.7.4 Hyperbolic mesh generation 37
2.8 Conclusion 39

CHAPTER 3 LEVEL-SET ICE ACCRETION 41
3.1 Context 41
3.1.1 Implicit interface-capturing 42
3.2 Level-set equation 43
3.2.1 Flux computations 45
3.2.2 Time integration 46
3.2.3 Boundary conditions 47
3.2.4 Eikonal equation 48
3.2.5 Re-initialization equation 49
3.3 Icing velocity definition 50
3.3.1 (Re)-Propagation equation 51
3.3.2 Boundary conditions 52
3.3.3 Impact of velocity type 52
3.4 Explicit tracking 53
3.5 Global conservative level-set process 54
3.6 Space-time accuracy study 55
3.7 Results 57

3.7.1 Manufactured Case . . . 57
3.7.2 Rime Ice . . . 59
3.7.3 Glaze Ice . . . 61
3.7.4 Error Analysis . . . 63
3.8 Conclusion . . . 64

CHAPTER 4 HYPERBOLIC ICE ACCRETION . . . 65
4.1 Rationale . . . 65
4.2 Numerical modelling . . . 67
4.2.1 Link with Eikonal solution . . . 70
4.3 Verification of accuracy . . . 73
4.4 Icing results . . . 75
4.4.1 Manufactured Case . . . 75
4.4.2 Rime Ice . . . 77
4.4.3 Glaze Ice . . . 79
4.5 Conclusion . . . 81

CHAPTER 5 STOCHASTIC ICE ACCRETION . . . 82
5.1 Literature . . . 82
5.2 Numerical modelling . . . 83
5.2.1 Global process . . . 84
5.2.2 Cartesian advancing front mesh . . . 84
5.2.3 Impingement probability . . . 86
5.2.4 Freezing probability . . . 87
5.2.5 Ice front extraction . . . 89
5.2.6 Pseudo-random number generator . . . 90
5.3 Cell size effect . . . 90
5.4 Icing results . . . 92
5.4.1 Ice shape averaging . . . 93
5.4.2 Rime Ice: Case 33 . . . 93
5.4.3 Rime Ice: Case 1 . . . 96
5.5 Conclusion . . . 98

CHAPTER 6 METHODS COMPARISON STUDY . . . 100
6.1 Background . . . 100
6.2 Quantitative metrics . . . 102
6.3 Qualitative criteria . . . 104

6.4 Experimental icing cases 105
6.4.1 Glaze Ice: Case 401 106
6.4.2 Rime Ice: Case 405 108
6.5 Discussion 111
6.6 Conclusion 114

CHAPTER 7 CONCLUSION 115
7.1 Advancement of knowledge 116
7.1.1 B-splines geometry 116
7.1.2 Level-set method 116
7.1.3 Hyperbolic method 117
7.1.4 Stochastic behaviour 117
7.1.5 Comparative study 117
7.2 Limitations 118
7.3 Recommendations 119

REFERENCES 120

APPENDICES 133

LIST OF TABLES

Table 3.1	Level-set cases parameters	57
Table 3.2	Rime case methods combination	59
Table 3.3	Ice mass error	63
Table 4.1	Rime case ice mass error	79
Table 4.2	Glaze case ice mass error	80
Table 5.1	Results of the stochastic cell size effect on density	92
Table 5.2	Stochastic icing cases parameters	93
Table 5.3	Stochastic density results for rime case 33	93
Table 5.4	Stochastic density results for rime case 1	97
Table 6.1	Icing cases parameters for the validation	106
Table 6.2	Glaze case 401 quantitative metric errors : Algebraic method	107
Table 6.3	Glaze case 401 quantitative metric errors : Tracked level-set method	109
Table 6.4	Glaze case 401 quantitative metric errors : Hyperbolic method	109
Table 6.5	Rime case 405 quantitative metric errors : Algebraic method	110
Table 6.6	Rime case 405 quantitative metric errors : Tracked level-set method	110
Table 6.7	Rime case 405 quantitative metric errors : Hyperbolic method	110
Table 6.8	Characteristics of the ice accretion methods	113
Table B.1	Quantitative metrics : Experiments	138
Table B.2	Glaze case 401 quantitative metrics : Algebraic method	138
Table B.3	Glaze case 401 quantitative metrics : Tracked level-set method	139
Table B.4	Glaze case 401 quantitative metrics : Hyperbolic method	139
Table B.5	Rime case 405 quantitative metrics : Algebraic method	139
Table B.6	Rime case 405 quantitative metrics : Tracked level-set method	140
Table B.7	Rime case 405 quantitative metrics : Hyperbolic method	140

LIST OF FIGURES

Figure 1.1 Examples of ice formation on aircraft. Photographs from (a) National Aeronautics and Space Administration (NASA) [64], (b) Potapczuk [76] and (c) Federal Aviation Administration (FAA) [30] 1

Figure 1.2 Examples of icing wind tunnel experiment results. Photographs by NASA [64] 2

Figure 1.3 Machined ice shape installed on aircraft for in-flight certification test. Figures adapted from Sadek [86] 4

Figure 1.4 Variability within experimental results over a NACA0012 airfoil. Data from Trontin *et al.* [100] 5

Figure 1.5 Airfoil ice morphologies 6

Figure 1.6 Micro-scale experimental and numerical simulations. Figures adapted from (a) Attarzadeh *et al.* [3] and (b) McClain *et al.* [57] 7

Figure 1.7 Ice crystal icing. Figures adapted from (a) Bansmer and Baumert [5] and (b) Norde [67] 8

Figure 1.8 Swept wing ice morphologies. Figures (b) to (d) adapted from Diebold *et al.* [28] 9

Figure 2.1 Structure of a multi-layer icing software 16

Figure 2.2 Flow fields around different airfoils 19

Figure 2.3 Particle trajectories 21

Figure 2.4 Droplet trajectory lines and impingement on a multi-element MDA airfoil 23

Figure 2.5 Control volume diagrams for mass and energy balance equations . . . 24

Figure 2.6 Examples of the thermodynamic model results on a glaze ice case over a NACA0012 airfoil 25

Figure 2.7 Examples of the geometry evolution methods. Figure 2.7(b) adapted from Thompson *et al.* [96] 27

Figure 2.8 Iterative algebraic method example 29

Figure 2.9 Close view of a coarse grid behind a ice horn for different generation techniques 30

Figure 2.10 Quality of a grid generated with the elliptic method on a ice horn geometry 31

Figure 2.11 A cubic B-spline built with nine control points and thirteen knots. Figure adapted from Bourgault-Côté *et al.* [16] 32

Figure 2.12 Cubic B-spline reconstructions for two different data sets using (a) 27 and (b) 45 control points. Figures adapted from Bourgault-Côté *et al.* [16] 34
Figure 2.13 Effect of the approximation tolerance on the B-spline curve 35
Figure 2.14 Comparison between the approximation method of Algorithm 1 to the fitting method from OpenCascade on an experimental ice shape (GLC305 airfoil) 36
Figure 2.15 Main steps of the curvature adaptive one-dimensional (1D) discretization method 38
Figure 2.16 Quality of a grid generated with the hyperbolic method on a ice horn geometry 39
Figure 3.1 Examples of issues caused by an explicit surface mesh deformation in convex and concave regions 42
Figure 3.2 Front merging process of the level-set equation 43
Figure 3.3 Flux computation upwind schemes 45
Figure 3.4 Ghost cell filling through extrapolation method for two scenarios . . . 47
Figure 3.5 Impact of the spatial accuracy on the Eikonal solution 48
Figure 3.6 Impact of the icing velocity formulation on the velocity and icing solutions 52
Figure 3.7 Main steps of the explicit tracking process 54
Figure 3.8 Conservative level-set process 55
Figure 3.9 Results of the (a) spatial and (b) temporal accuracy order verification 56
Figure 3.10 Manufactured results for level-set and algebraic methods. Figures adapted from Bourgault-Côté *et al.* [16] 58
Figure 3.11 Results for the rime ice case "CASE_01" from Trontin *et al.* [100]. Figures adapted from Bourgault-Côté *et al.* [16] 60
Figure 3.12 Results for the glaze ice case "Condition No. 3" from Trontin *et al.* [100]. Figures adapted from Bourgault-Côté *et al.* [16] 62
Figure 4.1 Ice-like features of the hyperbolic mesh generation method (coarsened for display) 67
Figure 4.2 Examples of ice accretion results from the hyperbolic ice accretion solver 69
Figure 4.3 Limiting cases for the hyperbolic icing method 70
Figure 4.4 Local effect of the modified hyperbolic source term 72
Figure 4.5 Effect of Eikonal gradient deviation source term in concave region . . 73
Figure 4.6 Flat plate canonical icing results 74
Figure 4.7 Results of the sub-layers order of accuracy study 75
Figure 4.8 Manufactured results for hyperbolic and algebraic methods 76

Figure 4.9	Hyperbolic icing solver results for the NACA0012 rime case 1	78
Figure 4.10	Hyperbolic icing solver results for the NACA0012 glaze condition 3	80
Figure 5.1	Process of the stochastic ice accretion	85
Figure 5.2	Advancing front organization	86
Figure 5.3	Comparison between predicted and stochastic impingement water mass. Figure adapted from Bourgault-Côté and Laurendeau [12]	87
Figure 5.4	Freezing process in the stochastic ice accretion	88
Figure 5.5	Stochastic ice solution before and after the ice front extraction	89
Figure 5.6	Cell size effect on the stochastic ice accretion	91
Figure 5.7	Results of the stochastic cell size convergence study	92
Figure 5.8	Results of the stochastic icing convergence in layers study for rime case 33	94
Figure 5.9	Stochastic icing solver results for the NACA0012 rime case 33	94
Figure 5.10	Comparison of the averaged stochastic ice shapes for case 33	95
Figure 5.11	Stochastic icing solver results for the NACA0012 rime case 1	96
Figure 5.12	Comparison of the averaged stochastic ice shapes for case 1	97
Figure 6.1	Ice horn metrics and ice thickness distribution	103
Figure 6.2	Ice coverage and ice sectional area metrics	104
Figure 6.3	General ice shape qualitative evaluation	105
Figure 6.4	Icing results for the NACA0012 glaze case 401	108
Figure 6.5	Icing results for the NACA0012 rime case 405	111
Figure A.1	Hyperbolic mesh generation example	136
Figure A.2	Hyperbolic mesh quality metrics	137

LIST OF SYMBOLS AND ABBREVIATIONS

c	Reference chord length	m
$\mathbf{d}$	Non-dimensional distance vector	-
f_{wall}	Freezing fraction at wall	-
$\vec{g}$	Gravitational acceleration	m/s^2
h_{ice}	Predicted ice height	-
$h_{ice}^{l_{sub}}$	Ice height at an algebraic sub-layer iteration	-
k_s	Equivalent sand grain roughness	m
l_{sub}	Algebraic sub-layer iteration	-
$\dot{m}$	Mass flow rate	kg/sm^2
$\hat{\mathbf{n}}$	Plane unit normal	-
$\mathbf{n_f}$	Facet unit normal vector	-
$\mathbf{n}_\phi$	Interface normal vector	m^2
$\vec{n}$	Grid unit normal	-
N_k	Curve knots multiplicity	-
N_{CP}	Number of control points in a curve	-
N_{sub}	Algebraic number of sub-layers	-
p	Curve degree	-
P_{fr}	Freezing probability	-
$\mathbf{r}$	Non-dimensional node coordinate relative to origin	-
s	Curvilinear distance	m
t	Physical time	s
T_s	Surface temperature	K
T_{freeze}	Water freezing point temperature	K
$\mathbf{u_d}$	Non-dimensional droplet velocity vector	-
$\vec{U}$	Velocity vector	m/s
$\mathbf{V_i}$	Ice growth velocity field	m/s
$\mathbf{V}$	Interface velocity	m/s
(x, y)	Cartesian coordinates	-

Subscripts

η	η-direction related derivative.
ξ	ξ-direction related derivative.

a	Air related variable
$cond$	Conduction
$conv$	Convection
d	Droplet related variable
ev	Evaporation
f	Facet-centre value.
H	Neighbour value.
HL	Highlight
I	Cell-centre value.
i	Ice related variable
ice	Ice related variable
imp	Impingement
in	Inflow
kin	Kinetic
L	Left state
$local$	Variable evaluated locally
out	Outflow
R	Right state
rad	Radiation
st	Stagnant
su	Sublimation
w	Water related variable
$wall$	Variable evaluated at the nearest wall

Greek Symbols

α	Water volume fraction	-
α_ϕ	Relaxation factor for hyperbolic icing source term	-
β	Droplet collection efficiency	-
ΔS	Facet area	m^2
Δs	Linear facet length on the surface mesh	-
$\Delta s^{l_{sub}}$	Corrected linear facet length for an algebraic sub-layer iteration	-
Δs_{cell}	Stochastic cell size	m
ΔV	Cell volume (area in 2D)	-
δ_h	Ice horn length	m
δ_{ice}	Ice thickness	m
ϵ	Curve global L2 error to the data points	-

ϵ_Δ	Local improvement tolerance for a knot insertion	-
ϵ_{curve}	Global curve approximation tolerance	-
ϵ_{loc}	Curve error to the data points in a local knot span	-
γ	Growth rate for curve discretization	-
λ_{max}	Maximum eigenvalue	-
Ω	Volume swept by the interface	m^3
Ω_{ice}	Ice volume to achieve (predicted by thermodynamic solver)	m^3
ϕ	Non-dimensional level-set signed distance	m
ρ	Density	kg/m^3
ρ_{ice}	Ice density	kg/m^3
τ	Pseudo-time	-
θ_h	Ice horn angle	$^\circ$
ε	Local cell characteristic length	m
φ	Propagation variable	-

Abbreviations

AoA	Angle of Attack	$^\circ$
CAD	Computer Aided Design	
CFL	Courant-Friedrichs-Lewy	
CPU	Central Processing Unit	
FAA	Federal Aviation Administration	
FAR	Federal Airworthiness Regulations	
FVM	Finite Volume Method	
HPC	High Performance Computers	
HTC	Heat Transfer Coefficient	
ICI	Ice Crystal Icing	
IRT	Icing Research Tunnel	
LWC	Liquid Water Content	kg/m^3
MDO	Multidisciplinary Design Optimization	
MVD	Median Volume Diameter	m
NASA	National Aeronautics and Space Administration	
NURBS	Non-Uniform Rational B-Spline	
PDE	Partial Differential Equation	
PRN	Pseudo-Random Number	
RANS	Reynolds Averaged Navier-Stokes	

RMS	Root Mean Square	
SLD	Supercooled Large Droplets	
SSD	Supercooled Standard Droplets	
SWIM	Shallow Water Icing Model	
TFI	Transfinite Interpolation	
TWC	Total Water Content	kg/m^3

LIST OF APPENDIXES

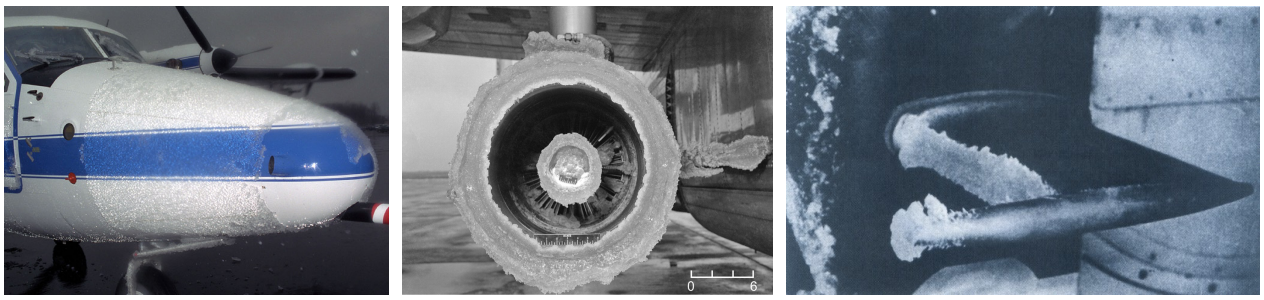
APPENDIX A HYPERBOLIC MESH GENERATION 133
APPENDIX B COMPARATIVE STUDY DATA 138

CHAPTER 1 INTRODUCTION

Icing is a hazardous natural phenomenon which has always been a challenge for aviators since the first powered flight passed through a cloud. It is the known cause for multiple aircraft incidents through history even though regulations are meant to be strict enough to ensure aircraft protection. This chapter presents an introduction on the subject and follows with a description of the state-of-the-art in numerical icing simulation. More specifically, ice accretion modelling is discussed, and finally, the research objectives of this project are defined.

1.1 Aircraft icing

The formation of ice on aircraft is a problem known to pilots since the beginning of aviation and the first documented encounters and fatalities due to ice date back to the mid-1920s, when the pioneers of flying by instruments could no longer avoid clouds in their flights to deliver mail. Since then, research in icing prediction has always been directed toward flight safety, with a succession of up and downs which could probably be correlated with fatal incidents and false victory declarations over the issue [55]. Ice can accumulate not only on aircraft main components, such as the wings, the nose or the engines, but also on or in the flight instruments, such as Pitot probes or antennas. Figure 1.1 shows good examples of such ice accumulations.

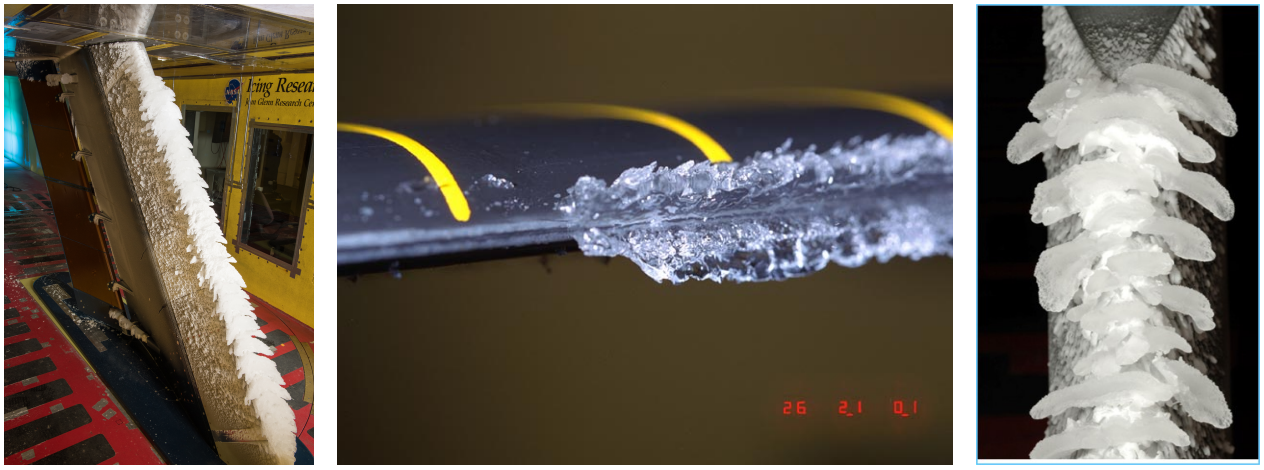


(a) Ice on a nose cone (b) Ice on an engine (c) Ice on a Pitot probe

Figure 1.1 Examples of ice formation on aircraft. Photographs from (a) NASA [64], (b) Potapczuk [76] and (c) FAA [30]

To better understand icing, many extensive data acquisition studies have been performed in the past decades, such as the one produced with the NASA Icing Research Tunnel (IRT), which is the first wind tunnel to be built for icing research [75]. Figure 1.2 shows some

examples of icing experiments obtained in the IRT facility through the years. The purpose of these experiments concerns the prediction of aerodynamic performance degradation due to icing and the design of more efficient de-icing systems for aircraft. Until the early 1990s, the research was mainly done for icing conditions in Supercooled Standard Droplets (SSD) conditions, which represent clouds of droplets at a temperature below the freezing point.



(a) Iced swept wing model (b) Clear ice with horns (c) Ice scallops

Figure 1.2 Examples of icing wind tunnel experiment results. Photographs by NASA [64]

Ice accretion such as seen in Figures 1.1 and 1.2 is usually divided into three categories: i) rime ice, ii) glaze ice and iii) mixed ice. Rime ice is characterized by low temperature conditions with low Total Water Content (TWC). It is opaque and fragile and it is generally accumulated in a thin smooth shape, like frost. The opacity comes from the capture of air in the ice at the impact on a solid surface. On the other hand, glaze ice occurs at warmer temperature and when the TWC is higher. Glaze is often clearer and denser than rime ice as it is formed by a water film which flows on the solid surface before freezing. This water film, called runback water, is the reason why glaze ice shapes can be much more complex than rime ice shapes, featuring horns or scallops such as shown on Figures 1.2(a) to 1.2(c). Finally, mixed ice combines both types where the glaze ice is generally located around the stagnation points area only, with the remaining ice being rime. The TWC is found by summing the amount of liquid water (droplets) and solid water (crystals) in the air. The distribution between both dispersed phases depends on atmospheric conditions in part, but mostly on the altitude. Liquid Water Content (LWC) is used instead of the TWC when only the droplet phase is considered.

A majority of ongoing research in icing now focuses on two main topics : i) icing due to Supercooled Large Droplets (SLD) and ii) Ice Crystal Icing (ICI) simulation. SLD represents the collective conditions found in freezing drizzle and freezing rain, mainly because large

droplets can be found in such conditions with diameters beyond $1000\mu m$. An encounter with a cloud of SLD generates completely different ice shapes compared to SSD ice accretion and can cause a more serious degradation of aerodynamic performances [55]. On the other hand, ice crystals are solid particles of ice presenting a large variety of shapes which can be found mostly at high altitude. ICI is even less understood than droplet icing as ice crystals can stick to a surface and generate shapes where erosion plays an important role. Such crystals are known to be the cause of multiple engine failures even when icing protection systems are active as they can agglomerate on non-protected parts of the engines [20].

1.1.1 Regulations and certification

An aircraft flies in icing conditions regularly in its life and it is necessary for the manufacturers to consider icing protection in their design. Among all the regulations related to aircraft, icing protection requirements were added in two distinct phases by the Federal Aviation Administration (FAA). First, the Appendix C of Federal Airworthiness Regulations (FAR) part 25 took effect in December 1964 and enforces the aircraft manufacturers to test their integrated icing protection for new aircraft considering different flight phases. It also defines SSD conditions as the icing envelope to be tested, as the droplet diameter in the tables and figures is limited to $50\mu m$ [31]. These regulations were modified recently due to several accidents that happened in the previous quarter-century.

As a first example, the disastrous accident of an ATR-72 on October 31, 1994, near Roselawn, Indiana, caused 68 deaths and brought the focus on icing as it was stated to be the probable cause of the accident. Furthermore, the accident investigation report from the American National Transportation Safety Board (NTSB) highlights i) that the airplane had encountered icing conditions with droplets having diameters from $100\mu m$ to $2000\mu m$, which correspond to freezing rain conditions and ii) that the certification authorities should include these conditions in the regulations [66].

Another major accident happened in 2009 on June 1^{st} when an Airbus A330-203, with 228 people on board, crashed in the Atlantic Ocean after taking off from Rio de Janeiro. The French Bureau d'Enquêtes et d'Analyses (BEA) investigated the crash and reported that the cause of the crash is the loss of airspeed indication due to ice crystals obstructing the Pitot probes. Furthermore, the report recommended undertaking studies to ascertain cloud compositions relatively to ice crystals and to modify certification requirements according to these results [21].

These two accidents, among others, led to the development of Appendix O [32] and D [33] in FAR parts 25 and 33, respectively, which were applied in November 2014. The former

introduces the need for aircraft manufacturers to certify new aircraft for SLD conditions while the latter concerns ice crystals.

(a) Machined ice shape on wing leading edge

(b) Machined ice shape on the nose

Figure 1.3 Machined ice shape installed on aircraft for in-flight certification test. Figures adapted from Sadek [86]

The certification process requires the manufacturers to demonstrate the flight capability of their new aircraft in these icing conditions, as it was done for the conditions stated in Appendix C of FAR part 25. To do so, the manufacturers add artificial ice shapes on the wings to reproduce specific ice conditions obtained when the anti- or de-icing devices are turned off, like shown in Figure 1.3. This is necessary for aircraft and crew safety as natural icing conditions are difficult to predict. To obtain these artificial shapes, numerical simulations or icing wind tunnel experiments are used, but the latter are expensive and restrictive. Indeed, matching the cases dimensionless numbers with experiments, such as the Mach or Reynolds numbers and the ratio between the water droplets size and the leading edge radius, requires to design alternative airfoil shapes. The goal is to match the leading edge radius to droplets ratio within an airfoil small enough to fit in the wind tunnel sections and results in the flow field matching only in the leading edge region [38, 77]. Furthermore, icing experiments are known to be highly variable from one experimental test to another, even within the same icing wind tunnel and with the same parameters, as it can be seen on Figure 1.4 where multiple experimental results obtained on a NACA0012 airfoil [100] are showing a high variation over the ice thickness and ice horns lengths and locations. Therefore, ice accretion simulation software are now being improved and validated by wind tunnel experiments to reduce the need for wind-tunnel experiments within the certification process. The SLD and ICI conditions represent phenomena which are still not well understood, to say the least, and their numerical simulation requires a higher comprehension on the geometry physics leading

to complex shapes such as glaze ice scallops [28], but also on the modelling of more complex physics phenomena, such as droplets breakup.

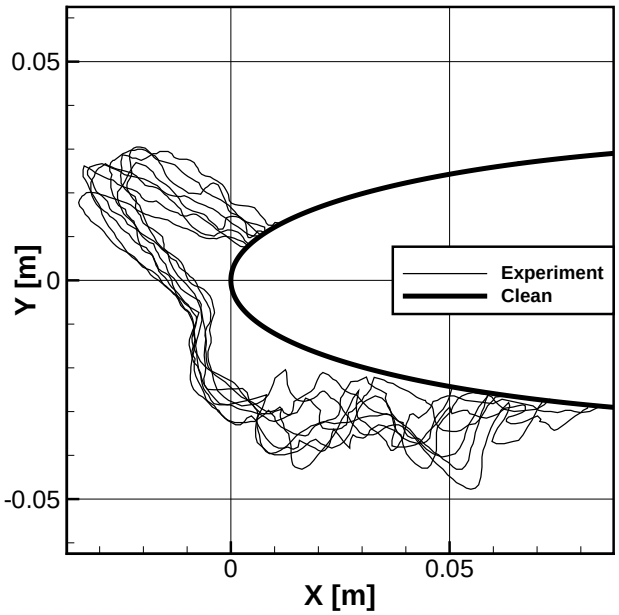


Figure 1.4 Variability within experimental results over a NACA0012 airfoil. Data from Trontin *et al.* [100]

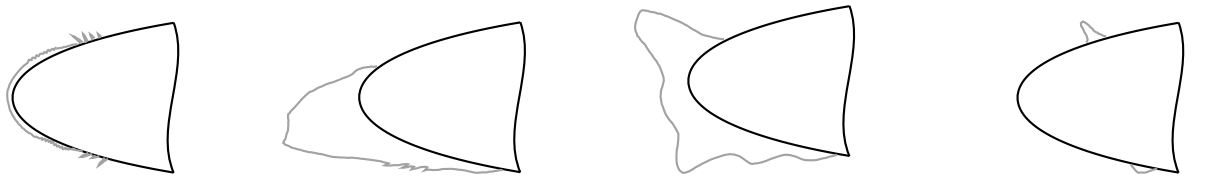
1.2 Ice morphologies

Icing covers a large variety of physical phenomena, from the impact of droplets or ice crystals on cold surfaces to the solidification process itself at different scales. Research efforts are therefore concentrated on specific types of ice accretion, or specific phenomena related to ice accretion. Until the early 1990s, ice shapes were studied in two dimensions most of the time and led to the maturation of knowledge on airfoil ice and to the creation of large databases of two-dimensional (2D) experimental data. Since then, research groups around the world are now focusing on understanding the icing phenomenon at a micro-scale, such as roughness accretion or single droplet solidification, and at macro scale for three-dimensional (3D) accretions like for ice crystals accretion or icing on swept wings. These different categories of ice morphologies which are being researched are described briefly below.

1.2.1 Airfoil ice

Ice on 2D airfoils can be separated in categories depending on the ice geometry and on its effects on the aerodynamic performances: i) roughness, ii) streamwise, iii) horn and iv) spanwise ridge [18]. Figure 1.5 present an example of ice shape for each of these categories

with increasing negative effects on aerodynamic performances from left to right.



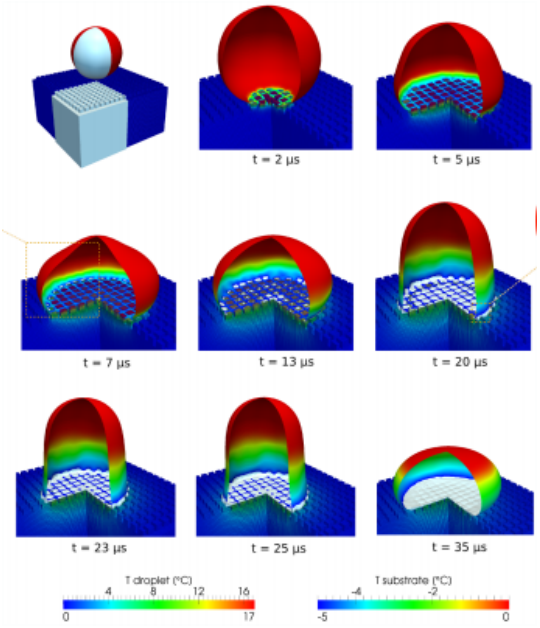
(a) Roughness ice (rime) (b) Streamwise ice (rime) (c) Horn ice (glaze/mixed) (d) Spanwise ridge ice (glaze)

Figure 1.5 Airfoil ice morphologies

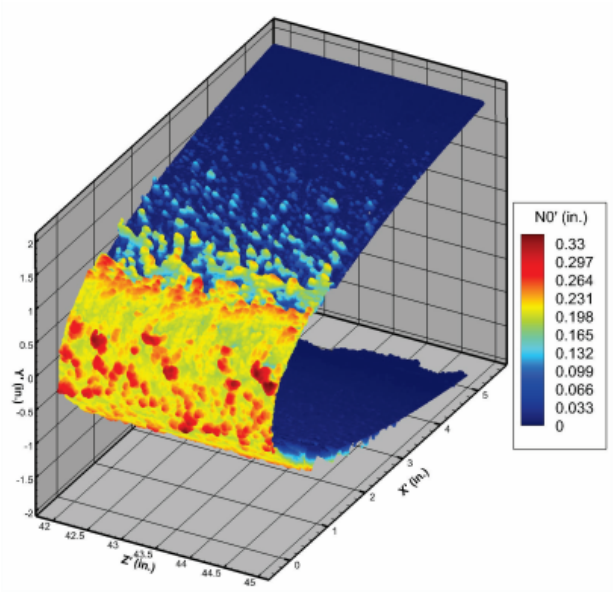
As seen on Figure 1.5(a), roughness ice is made of a thin layer of rime ice in the leading edge region, which is smooth near the stagnation point and becomes rougher in the end. Furthermore, formations called rime feathers are present behind the ice layer region. Overall, this ice type causes a viscous drag increase due to the roughness and a lower maximum lift coefficient [18]. Streamwise ice (Figure 1.5(b)) is formed mostly by water freezing at the impact on a cold surface, forming an agglomerate which does not have significant effects on the airflow. On the other hand, horn ice (Figure 1.5(c)) and spanwise ridge ice (Figure 1.5(d)) can considerably affect the aerodynamic performances of an airfoil, by increasing the pressure drag due to large separation zones behind the ice formations. The first is formed generally by glaze ice or mixed ice, when a film of runback water is present on the surface of the ice and freezes while it is flowing away from the stagnation region, creating horns of ice. The second is obtained in the same conditions as horn ice, on a surface with a protected region around the leading edge where the water cannot freeze, but can still flow up to an unprotected zone where it freezes in the shape of a ridge.

1.2.2 Surface roughness and micro-scale

Many studies have been performed experimentally and numerically to better understand the micro-scale solidification process of water. Fortin *et al.* [34] extensively studied the forces applied on liquid droplets moving on a surface and the freezing process of runback water in order to propose an analytic model for surface roughness caused by ice. On a similar topic, Attarzadeh *et al.* [3] recently presented numerical analysis of a droplet impacting a cold surface modelled with hexahedral shapes to simulate roughness in a microscopic scale, as seen on Figure 1.6(a). The purpose was to understand how the roughness characteristics affect the freezing of water droplets on a surface. On a larger scale, new techniques are being developed by NASA for experiments on ice roughness to visualize the ice roughness and its micro-scale shape, as seen on Figure 1.6(b) where a topography of the roughness thickness is shown [57].



(a) Modelling of droplet freezing on a rough surface



(b) Topography of ice roughness thickness

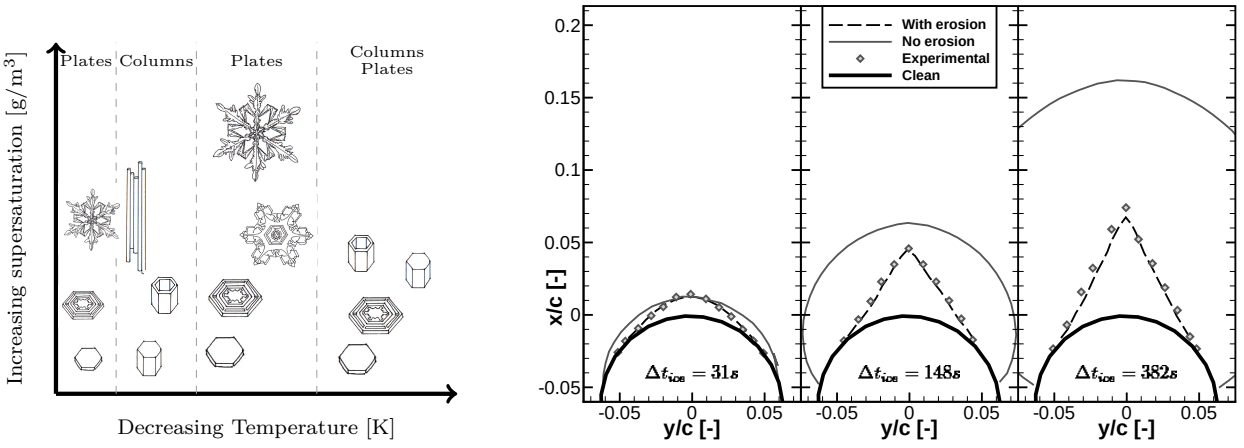
Figure 1.6 Micro-scale experimental and numerical simulations. Figures adapted from (a) Attarzadeh *et al.* [3] and (b) McClain *et al.* [57]

1.2.3 Ice crystal agglomeration

Ice shapes obtained with crystals resemble streamwise rime ice shapes, with additional phenomena playing an important role : i) erosion and ii) sticking of particles. ICI is not a well-known phenomenon, even when compared to icing in general, and the shape of the crystals affects how the ice is agglomerated on the surfaces. Bansmer and Baumert [5] presents a classification of such crystal shapes as spheroids, columns, plates, rosettes and irregular shapes and some examples are presented in Figure 1.7(a). Droplets are subject to rebound and fragmentation when hitting a surface, but ice crystals can also stick directly to the surface without flowing on it. Furthermore, as the ice is an agglomeration of particles, a part of the crystals are removed by erosion, resulting in a sharp ice shape as the one seen on Figure 1.7(b) where the erosion is modelled with increasing number of steps and is compared by Norde [67] to experimental results. This shows how the intensity of the erosion affects the global shape of the ice, which requires a modelling technique able to handle sharp edges.

1.2.4 Swept-wing ice

The focus of this research project is on icing caused by liquid water droplets on highly 3D surfaces and in particular on complex ice shapes generated in SLD conditions. These



(a) Ice crystal morphologies

(b) Ice crystal accretion on a cylinder

Figure 1.7 Ice crystal icing. Figures adapted from (a) Bansmer and Baumert [5] and (b) Norde [67]

critical shapes are encountered on swept wings, where the airflow has 3D characteristics and requires a further classification depending on their accumulation shape and conditions. This is highlighted by the work of Diebold *et al.* [28] who presented a review of actual knowledge on swept-wing icing and suggested such an ice shape classification system based on the naming used in airfoil ice. Note that the ice categories are noted to be nominally 3D when the produced ice shapes are smoothly varying in the spanwise direction, comparatively to highly 3D categories where it would not be the case. The different categories are listed below in approximate ascending order of aerodynamic performance degradation.

- Ice roughness: On wings with a high sweep angle, the surface roughness caused by the deposit of ice is different than on airfoils, notably by the absence of a smooth zone around the leading edge, although this is not always the case as seen in recent experiments from [57].

- Streamwise ice : Formation seen generally in rime ice conditions where the ice accumulates in the leading edge area in an agglomerated shape, which prolongs the wing shape in the streamwise direction, as seen on Figure 1.8(a).

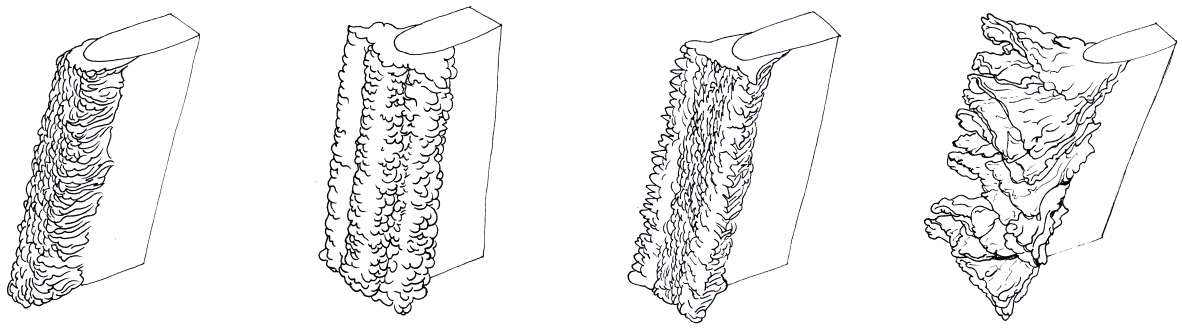
- Horn ice: As airfoil ice, swept-wing icing can produce horns of ice on the wing and the shape of these horns depends on the sweep angle and on the icing conditions. This category can be further subdivided into three stages depending on the effect on the aerodynamic flow.

  – No scallops: Similar to a 2D ice cross-section which would have been extruded

along the spanwise direction, as seen on Figure 1.8(b), this category of ice is nominally 3D and the effects on the flow are mainly on streamwise attributes, such as the presence of vortices around the leading edge.

  - Scallops tips: At this stage, scallops are starting to appear, as seen on Figure 1.8(c), but without perturbing the flow, which stays nominally 3D.
  - Complete scallops: In contrary to the two other sub-categories, the presence of fully developed scallops, such as can be seen on Figure 1.8(d), perturb the flows to the point of having high variations in the pressure distribution between two close locations in the spanwise direction. Therefore, this kind of shape can be considered highly 3D.

- Spanwise ridge ice: This formation of ice is obtained when the leading edge of a wing is protected against ice, but that ice formed a ridge farther downstream on the wing where there is no protection. This type of ice is especially dangerous by causing high aerodynamic performance degradations and is often related to SLD conditions.



(a) Streamwise ice on a swept wing

(b) Horn ice without scallops

(c) Horn ice with incomplete scallops

(d) Horn ice with full scallops

Figure 1.8 Swept wing ice morphologies. Figures (b) to (d) adapted from Diebold *et al.* [28]

1.3 Icing numerical modelling

Tools for numerical prediction of ice accretion play an important role in aircraft certification, as mentioned in §1.1.1. These tools typically comprise five main modules which run in a loop where the iteration counter is equivalent to a temporal counter as the ice is built up layer by layer. This is based on the hypothesis that the ice accretion process can be simulated as a quasi-steady phenomenon, where every module is run in steady state and only the geometry evolution itself is affected by the icing simulation time. The different modules are listed below in order of execution in an ice accretion software.

1. Volume field discretization from a geometry (mesh)
2. Continuous phase resolution (airflow)
3. Dispersed phase resolution (droplets and/or ice crystals trajectories)
4. Ice accretion on the surface (thermodynamic mass and energy exchanges)
5. Geometry evolution (new geometry from ice front displacement)
    5.1. Ice front displacement
    5.2. Geometry update

As the regulations are becoming stricter, great efforts are made in academia research groups to develop new algorithms to accurately model icing. Methods based on potential flow, Lagrangian particle tracking and algebraic accretion modelling were the first to be developed and are still being used by the aeronautical industry in the aircraft certification process. These tools, like CANICE [19] developed at Polytechnique Montreal in the 1990s for Bombardier, use older technology, but are extensively validated and certified for aircraft icing and can be considered the industrial state-of-the-art in icing numerical simulations. Other such tools include LEWICE 2.0 from NASA and TRAJICE 2 from Defence Evaluation and Research Agency (DERA) amongst the most well-known codes [108].

In comparison, the work of Bourgault *et al.* [9] and Beaugendre [7] introduced Partial Differential Equation (PDE) in icing in the form of Eulerian droplet trajectory solvers and aerodynamic solvers based on RANS equations and it has been the common baseline for most of the research work in ice accretion modelling ever since. The capacities to model correctly SLD or ICI is also a main research topic, as the physics of these phenomena is difficult to simulate numerically. Tools that can model these conditions are for instance : LEWICE 3.0 [105] developed by NASA, ONICE 2D/3D [104] and IGLOO 2D [99] developed by ONERA and FENSAP-ICE developed by Newmerical Technologies International [47].

Modules 2 to 4 have been extensively studied and improved in the last decades and are described in more details in Chapter 2. Meshing technologies for icing have also been developed to a satisfactory level for structured grids [45, 103], but unstructured meshes probably present the best approach for higher quality domain discretizations on full-aircraft configurations [107]. It would allow for example to perform mesh adaptation for icing [9] or mesh deformation when adding ice layers [98]. However, the geometry evolution component, which defines how the ice front is displaced before converting it into a new geometry, is not much referred to in the literature. Furthermore, a geometry deformation strongly depends on its representation format, but the concept of geometry in icing is generally not clearly defined

from a numerical point of view and modules 5.1 and 5.2 are often indiscernible from each other. Also, it is important to note that enabling multi-layer icing is a key feature for a geometry support as well as for the ice evolution method, because splitting the icing process in multiple layers allow to introduce the effect of the geometry on the physical phenomena modules, namely the continuous phase, the dispersed phase and the surface mass and energy exchanges. As such, it reduces the effect of the quasi-steady assumption and yields more accurate simulations. However, multi-layer simulations are challenging for different reasons :

- Risk of generating an invalid geometry;
- Risk of obtaining an invalid mesh due to high concavity in the geometry;
- Non-conservation of accumulated ice between layers;
- Necessity to run the process in an automated loop so that the output of the final module becomes the input of the first module.

In fact, the automation of the multi-layer approach leads to developing methods that aim to be 100% robust, which explains the difficulty due to the impossibility to reach such a level of robustness.

1.3.1 Geometry Representation

In Computational Fluid Dynamics (CFD) in general, a surface mesh is often mistaken for the geometry as CFD software generally know nothing about the true geometrical representation which is being solved on. Moreover, using a linear or bilinear surface representation as a geometry is a common practice that should be avoided as it might exhibit overly dense elements and slope discontinuities [26]. It is surprising that the Computer Aided Design (CAD) is not the standard baseline on which CFD tools are built, as discrete surfaces are preferred, even though engineering design is going toward more integrated tools in the Multidisciplinary Design Optimization (MDO) process. It led to the 1^{st} AIAA Geometry and Mesh Generation Workshop [26], which general objectives were to assess the actual state-of-the-art in geometry processing and its use in mesh generation and aerodynamic simulation software. The workshop highlighted many issues related to the geometry usage in CFD, for instance : i) simple geometries modelled with too many control points leading to undulations, ii) presence of geometry foldings, iii) conversion of strong geometry formats to known weaker formats for meshing, and iv) the use of non-manifold geometries.

In the context of icing, the geometry representations found in the literature are mostly of discrete type. 2D tools generally use linear facets as a support for the initial clean geometry,

but also for the ice front evolution and the update of the new geometry [78]. One exception found in the literature is a study by Thompson and Soni [97] with LEWICE 2D where Non-Uniform Rational B-Spline (NURBS) are used to redistribute the points on the clean or iced airfoils during the multi-layer process before proceeding to the meshing of the volume. In another case, Pueyo [78] uses B-splines as a smoothing support to redistribute the surface mesh points in the sub-iterations of the ice accretion process. In the case of 3D tools, it can be assumed that a Computer Aided Design (CAD) representation is used to generate the initial mesh, due to the complexity of the geometry. However, the literature suggests that the ice accretion software have only the knowledge of the surface grids, with discrete triangular or quadrilateral facets, during a multi-layer process [98]. This is mainly due to the complexity of generating a new smooth surface valid for simulation from an ice front interface. In any case, a continuous representation, such as splined curves (NURBS, B-splines, etc.), should be preferred to discrete faceting for new developments and better representation of the geometry characteristics in the software.

1.3.2 Ice evolution method

The standard method used to displace the ice front in 2D tools is often referred to as the algebraic method, also known as Lagrangian method. Nodes are explicitly displaced along a line, for instance the bisector line between two linear facets or the normal line from a facet, depending on an algebraic equation including the ice mass accretion rate [19]. Various improvements of this method, such as iterations, limiters or smoothing, have been created to control the front evolution and yield a better conservation of the ice volume [78]. However, it can produce an invalid surface grid with overlaps in concave regions and cannot be considered a robust approach. Furthermore, its extension to 3D is not straightforward. Thompson *et al.* [96] presented such an extension for unstructured surface grids, with limiters controlling the marching step normal to the surface.

Various other methods have been studied to replace the algebraic method. Pena *et al.* [71] presented a level-set approach to implicitly track the ice front interface within the 3D icing simulation modules in the NSMB software which operates on structured grids. The morphogenetic model from Szilder and Lozowski [93] is another different approach which accumulates ice discretely on a Cartesian lattice using a Lagrangian approach for the droplets impingement coupled to the use of probability functions for the freezing process and pseudo-random numbers. This method allows to generate complex glaze ice shapes, such as scallops on swept wing, which is promising. However, in both of these approaches, multi-layer capacity is not addressed.

1.4 Objectives of present study

The overarching objective of the present research is to obtain a set of appropriate methods for ice accretion modelling of complex ice shapes with a focus on the geometry definition and update phases into an automated and robust multi-layer process. As the project is supported by Bombardier in view of integration into their new icing simulation suite, a possible extension to 3D is of importance and efficiency is secondary. The developments for these methods are done in NSCODE-ICE, a multi-physics structured CFD tool which is developed at Polytechnique Montréal since 2012 [56, 11, 61]. The development line of thoughts in NSCODE-ICE is to use PDE-based models and algorithms to represent the various physical phenomena. In particular, some of the recent improvements in the icing modules are the study of 2.5D crossflow effects in icing [53] and the assessment of various thermodynamic modules [54]. These improvements also include a structured elliptic mesh smoother designed for ice shapes [45], although the generated grids' quality on glaze ice shapes featuring horns can further be improved. Considering that ice can grow in complex shapes as detailed earlier, especially in 3D, and that multi-layer simulations performed automatically involve the repetitive modification of the simulated geometry, the potential for simulation failures is high. This is even more the case if the grid quality deteriorates as the ice grows. Thus, the geometry evolution and the mesh generation modules are identified as limiting elements in the overall framework and in aero-icing in general. Therefore, specific objectives are proposed with a focus on geometry and its impact on mesh generation to achieve the main goal.

- Investigate algorithms enabling robust multi-layer ice accretion simulations for classical deterministic methods.
- Develop front evolution methods to model the ice accretion physics with ice mass conservation.
- Investigate the impact of adding a multi-layer technology into a stochastic ice accretion simulation.

In this work, the geometry is hypothesized as the critical core component that enables a robust multi-layer ice accretion process. The most challenging part of the project is to obtain an integrated in-CAD geometry system that would describe the details of an ice geometry. Mesh regeneration, as opposed to mesh movement, is chosen for the multi-layer icing process as it demonstrated robustness in previous studies [42]. However the meshing techniques need to be reviewed to improve the quality of the produced grids. Furthermore, the software

must be developed to operate on High Performance Computers (HPC), which increases the complexity as the interface between HPC and CAD is not trivial.

1.5 Thesis outline

Chapter 2 presents the multi-layer process in general. The techniques in use in NSCODE-ICE are briefly described in order to give a global comprehension of the software. The integration of a smooth geometry representation in the process is detailed. The next three chapters introduce three geometry evolution techniques and are followed by a study to compare the developed methods. Chapter 3 presents the developments of an implicit front-capturing method for ice accretion. An original method based on the resolution of hyperbolic equations is presented in Chapter 4 to propose a robust and computationally inexpensive explicit alternative for the evolution of the geometry under icing conditions. The introduction of stochastic behaviour in multi-layer icing modelling is then examined in Chapter 5. Chapter 6 follows with a comparison of the deterministic ice accretion methods with quantitative metrics and qualitative criteria over two NACA0012 experimental icing cases. Finally, Chapter 7 presents the conclusions and limitations of this study and some recommendations for future work.

CHAPTER 2 MULTI-LAYER ICING SIMULATION

The icing numerical simulation process was briefly presented in Chapter 1 and the challenges linked to the geometry update modelling were stated. The multi-layer simulation of icing is a challenging process where different modules are interacting. This chapter presents the overall multi-layer icing process, with a brief description of its different physical modules for completeness. The strategies proposed in this study for a higher robustness of the process are highlighted and described in more details.

2.1 Numerical simulation process

Icing is a combination of several natural phenomena interacting at different scales. A continuous phase, the air, transports a dispersed phase made of liquid droplets, or solid crystals, which then impacts a solid surface. This results in mass and energy transfers between the multi-phase fluid and the solid substrate. If the conditions are favourable, the liquid water deposited on the substrate freezes at a location and in a shape that depends on the global geometry, but also on the air flow, on the droplets chaotic dispersion and on the surface composition or roughness. This series of interactions between each physical phenomenon is difficult to represent entirely in a numerical framework and hypotheses need to be applied to simplify the modelling. Some of these hypotheses are presented later in this section.

Figure 2.1 shows the general structure of a multi-layer icing software. The process starts by defining a geometry and some conditions to simulate and then produce a volume mesh. The airflow around the geometry is solved to obtain an appropriate resolution of the boundary layer, as it drives the determination of the convective heat transfer between the air and the substrate. The water impingement rate on the surface is then obtained by solving the dispersed phase. Various sizes and types of particles can be modelled at this stage and the final water impingement rate is obtained through a weighted sum.

Next, the convective heat transfer from the air flow and the water impingement rate are used to solve the mass and energy balance equations on the surface, yielding an ice accretion rate mapping. The ice front is displaced, then extracted in the case of an implicit definition, and the final solution is converted into an ice geometry representation merged in the remains of the clean geometry untouched by icing. If the desired number of ice layers is not yet obtained, this new geometry is sent back to the mesh module and either a new mesh is generated or the previous mesh is deformed according to the surface evolution.

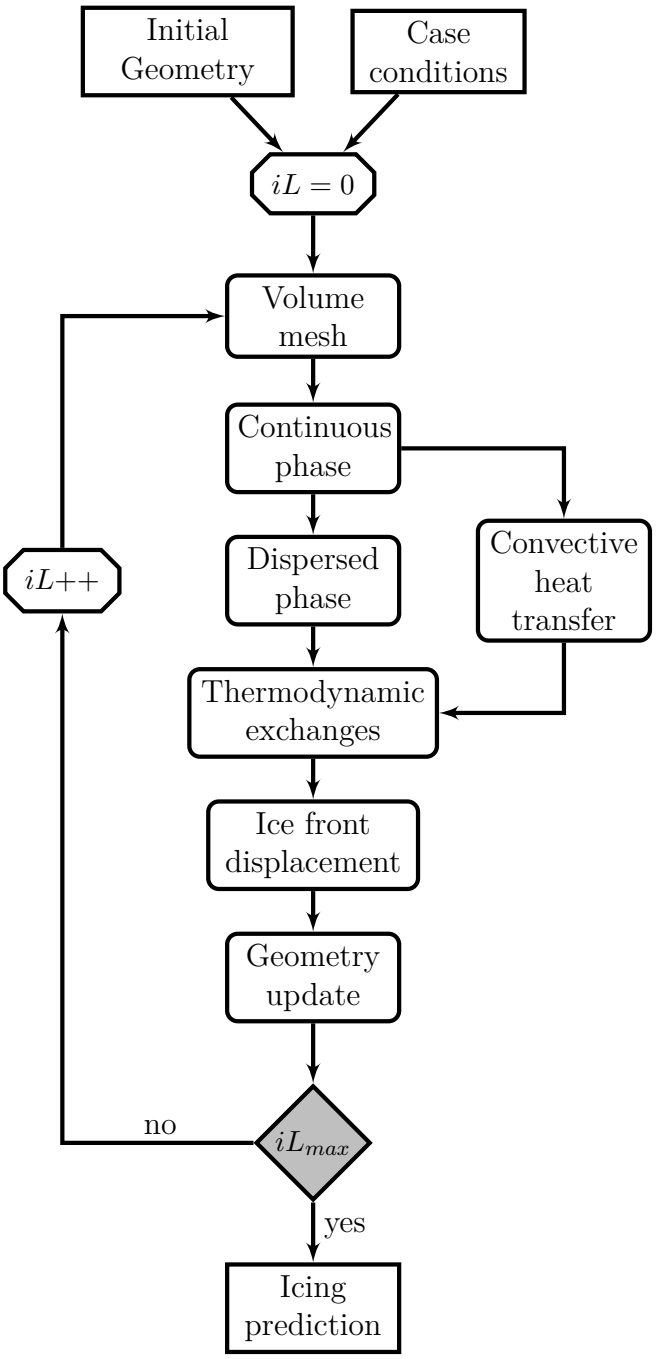


Figure 2.1 Structure of a multi-layer icing software

2.1.1 Simplifying hypothesis

The process described in Figure 2.1 is a simplified representation of the complete natural phenomenon where each module affects only the downstream components, with the exception of the geometry update part. This reduction of the modelling complexities of icing is made possible by stating the following hypothesis. These can be found at the basis of a large range of icing software, as well as in NSCODE-ICE. Some exceptions exist and are listed to the best of the author's knowledge.

The overall icing process is a quasi-steady phenomenon. This means that the main physical modules are solved in a steady state without considering unsteady parts of the equations. The global icing time scale is in the order of minutes, whereas an unsteady flow simulation requires a physical time step to be very small to represent adequately the unsteady effects, depending on the conditions and on the mesh. It is therefore not practical in realistic icing conditions and a multi-layer algorithm is preferred to represent the geometry time-dependent change during the computations [7]. In theory, splitting the global process in enough layers of ice should allow to converge toward a final ice shape [43] which are more or less similar to experimental results depending on the physical modelling accuracy. FENSAP-ICE contains an unsteady mode with which a fully unsteady icing simulation [2] or a quasi-unsteady icing simulation [36] can be computed, but it is resource-consuming and not practical for industrial applications.

The dispersed phase does not affect the continuous phase. Considering a small enough mass loading of the particles in the air, which is the case in icing conditions, the flow modelling equations do not need to be modified to take into account the presence of particles [9]. This hypothesis allows for a one-way coupling resolution of the multi-phase flow, as the particle trajectories are completely driven by the air flow and other external forces, while the air flow is not affected by the particles. Removing this hypothesis would require to solve the multi-phase flow in a coupled manner. This is the case in the study of Naterer [65] where the flow and droplets phases are solved together as well as with the heat transfer on the surface.

There is no interaction between the particles. These interactions represent mainly particles breakup, coalescence and collision [9]. The rationale justifying these simplifications is the computational resources required to handle such phenomena. As already mentioned, the mass loading of the particles in the air is low, meaning that the particles are sparse and interactions can be neglected without a loss of accuracy compared to the mesh discretization.

2.2 Continuous phase modelling

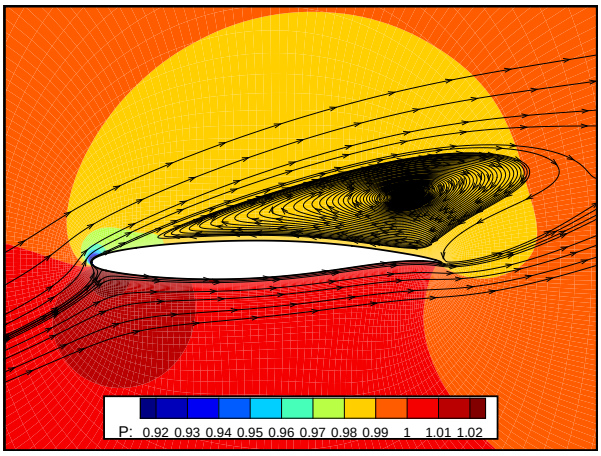
The aerodynamic flow field drives the particle trajectories in the volume domain and the heat transfer and the water runback water flow on the surface. It also allows to predict the aerodynamic performance degradation on the airfoil or aircraft due to ice contamination. In the first generation of icing software, such as CANICE [19] or LEWICE 2D [106], the aerodynamic inviscid flow over airfoils was computed with a potential method coupled to a boundary layer integration from the stagnation point. The main advantages of this approach are fast computation of the aerodynamic flow field and simplicity to implement as it required only a surface mesh of the geometry to be solved. However, a major drawback is that it cannot accurately predict aerodynamic performance degradations due to ice without being fully coupled to a boundary layer approach as it is limited by recirculation zones behind the ice horns. Nowadays, with computing resources being large enough, higher fidelity methods are being sought for the icing prediction leading to the integration of RANS solvers within icing codes [9]. This allows to solve complex geometries with a properly defined boundary layer in attached or separated states and for the consideration of compressible effects. One advantage is that the results can be used directly for aerodynamic performances degradation. The main drawback is that RANS equations need a high quality volume mesh to give quality results which introduces another complexity in the icing simulations.

2.2.1 RANS solver

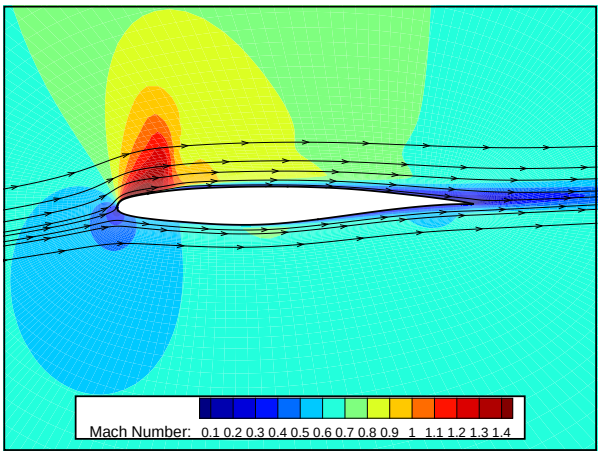
In NSCODE-ICE, the aerodynamic flow field is solved with a finite volume structured RANS method. The convective fluxes are discretized with a centered 2^{nd}-order spatial discretization scheme with scalar [49] or matrix [91] dissipation. The viscous term is using a 2^{nd}-order centered stencil. Several time stepping methodologies are available: hybrid multi-stage Runge-Kutta, semi-implicit Point-Jacobi or implicit LU-SGS [56]. The turbulence can be modelled with : i) Spalar-Allmaras 1-equation model [90], ii) SST $k-\omega$ 2-equations model [58] or iii) transitional $k-\omega-\gamma-Re_{\theta t}$ 4-equations model [61]. The flow solver also has the capacities to solve unsteady flows in time domain or frequency domain with various algorithms [74]. An original development from Ghasemi *et al.* [39] extended the capacity of the 2D flow solver in order to include the crossflow momentum equation in the resolution by applying the infinite swept wing hypothesis. More details about this method can be found in the work of Bourgault-Côté *et al.* [14] and its effects in icing are studied by Lavoie *et al.* [53].

Multiple acceleration techniques are used inside the flow solver, such as a full multigrid scheme allowing to spread numerical waves on coarser grids to accelerate their propagation up

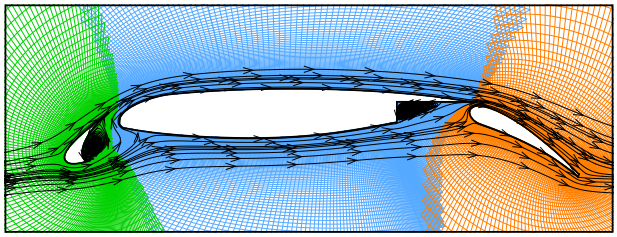
to the different grid physical and artificial boundaries. Furthermore, the solver is developed to support overset grids and contains a set of interpolation methods to pass information from one grid to the other. This particular capacity is useful in the context of this work to interpolate a scalar field at a specific ice front location and is thus described in more details in §2.6.2.



(a) Stalled supercritical airfoil



(b) Lovell airfoil in transonic flow



(c) Multi-element MDA airfoil on overset grid with streamlines

Figure 2.2 Flow fields around different airfoils

Figure 2.2 shows some examples of aerodynamic flow field solutions around airfoils. In particular, Figure 2.2(a) shows a supercritical airfoil passed its stalling point with a strong recirculation zone behind the airfoil, Figure 2.2(b) shows a Lovell airfoil with a shock in a transonic flow and Figure 2.2(c) shows a multi-element MDA airfoil solved using overset grids and highlighting recirculation zones in the slat and main element coves.

2.2.2 Convective heat transfer coefficient

One of the main components of the thermodynamic energy equation on the surface is the convective heat transfer, which is driven by the aerodynamic flow field. As the solver in NSCODE-ICE is a RANS solver, it is possible to directly extract the heat transfer on the

surface from the temperature gradient at the wall. It can then be used to compute a convective Heat Transfer Coefficient (HTC). However, this requires the use of an isothermal boundary condition which is different than the usual adiabatic wall condition. Two methods were implemented in NSCODE-ICE to use one flow solution [7] or two flow solutions [60] to compute the HTC. In the first method, Newton's law of cooling is applied with the difference between the imposed wall temperature and the farfield temperature. However, results show that the wall temperature choice affects the convective HTC. The second method uses two different flow solutions obtained with two wall temperatures to compute the HTC, resulting in a more accurate result than the first method which is less dependent on the wall temperature, but with a high additional cost.

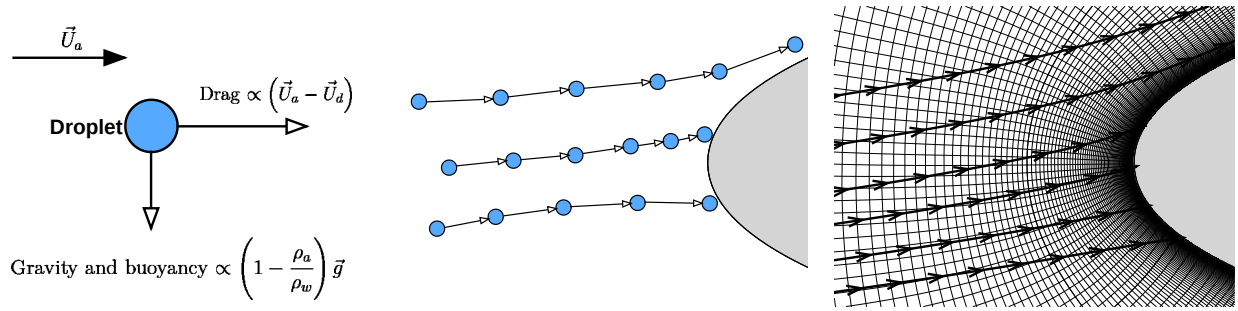
2.2.3 Surface roughness

The surface roughness height is one parameter that has a major impact on the results obtained in an icing simulation. On the one hand, it affects the skin friction which drives the runback water, while on the other hand the scale of the convective heat transfer depends in part on the roughness. As a consequence, the same case simulated with two roughness values can results in two very different ice shapes, e.g. by having ice horns growing at two different locations. However, the surface roughness due to ice is not well understood and experimental values cannot be measured when performing experiments in icing wind tunnels. In numerical simulations, the surface roughness due to ice is at best described analytically by semi-empirical models which depend on the case parameters and on the local characteristics of the water runback, as in the study of Fortin *et al.* [34]. Other methods used to define the roughness include constant values or chord-related values [80]. In NSCODE-ICE, the model of Shin *et al.* [88] depending on the icing case parameters is used to predict a constant value for an equivalent sand grain roughness (k_s) which is then applied on the airfoil surface as roughness height. Note that the roughness should be applied only on the ice, but it is instead applied everywhere due to the automation of the multi-layer icing process which limits how the surface boundary conditions are applied, including the roughness height definition.

2.3 Dispersed phase modelling

To obtain a water impingement mapping on the surface, the trajectories of liquid droplets and solid crystals must be modelled. According to the hypothesis of §2.1.1, these trajectories are driven by the aerodynamic flow without affecting it. This one-way coupling is done through the use of source terms as a drag force depending on the relative velocity between the air and the particles. The drag force also depends on the particle shape which is different for liquid

droplets and solid crystals [67]. Additional forces are applied on the particles without being as influential for in-flight icing, such as the gravity and buoyancy forces [29]. Nevertheless, these secondary forces are particularly important to include in the models for ground icing. The simplified forces decomposition on a particle in the air is shown in Figure 2.3(a).



(a) Close view of a particle and the forces acting on it

(b) Particles trajectories

(c) Field of particles

Figure 2.3 Particle trajectories

In the first generation of numerical icing simulations, the Lagrangian method was the only model used to compute these particle trajectories. The process requires to solve the well-known Newton's laws of motion for the particles velocity components and then to integrate the trajectories step by step. It is illustrated in Figure 2.3(b). This method is particularly efficient when potential methods are used for the aerodynamic flow field as volume grids are not required [19, 87]. However, the computational cost of this method increases with the size of the solid to simulate, and might become significant for a full-aircraft configuration. As the particles are seeded from a plane in front of the geometry, the seed area must be large enough and the seeding points must be dense enough so that the impingement area is completely covered with enough resolution. Furthermore, parallelization of such a method is complex to achieve efficiently if the domain is split among CPUs. The Lagrangian method is still being studied and different approaches have been proposed to solve some of these issues, like the recent study of Rendall and Allen [82] where the equations are solved in a finite volume way by using the volume grid generated for the aerodynamic flow simulation.

The pioneering work of Bourgault *et al.* [9] introduced in icing simulation an Eulerian model to model the trajectories of supercooled liquid droplets and their impact on an aircraft. By assuming that the particles are dense enough in the multi-phase flow, they can be modelled as a continuous flow with any numerical techniques allowing to solve a hyperbolic PDE with averaged values. The mesh generated for the RANS solver can be directly used, as seen in Figure 2.3(c), and the parallelization of the computations is simply an extension of any parallelism implemented in the aerodynamic flow solver. Furthermore, it simplifies the

simulation of complex geometries such as a full aircraft in deployed high-lift configuration by removing the need to follow a path line. However, some phenomena are more difficult to introduce in an Eulerian method than in a Lagrangian method, such as rebound, splashing or particles breakup, and research projects are dedicated to solve these issues [47, 50]. Also, simulating multiple particles size tremendously increases the computing cost, as each size or type must be simulated separately.

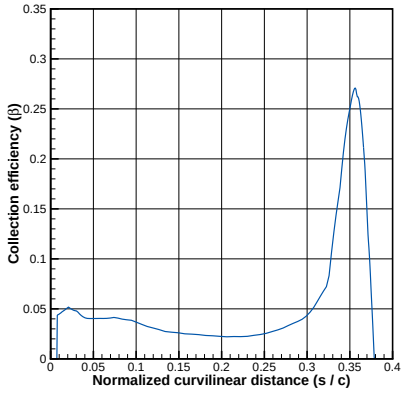
2.3.1 Eulerian droplet solver

The particle trajectory module in NSCODE-ICE solves the Eulerian droplet equations. A finite volume approach is chosen in order to reuse some technologies from the aerodynamic flow solver, such as the overset grids [17]. Furthermore, the concepts which led to the aerodynamic crossflow equation are applied to the w-momentum equation to obtain a droplet crossflow velocity allowing to solve the droplet trajectories in 2.5D on swept wings [11]. The equations are discretized with a 1^{st}-order upwind scheme and are solved with an implicit Euler scheme where the global system is solved with a BiCGStab Krylov-subspace iterative method [85]. The solver is simplified by solving each droplet equation (density and momentum in u, v and w) as a segregated scalar equation and this decoupling is made possible by the quasi-absence of off-diagonal terms in the system Jacobian.

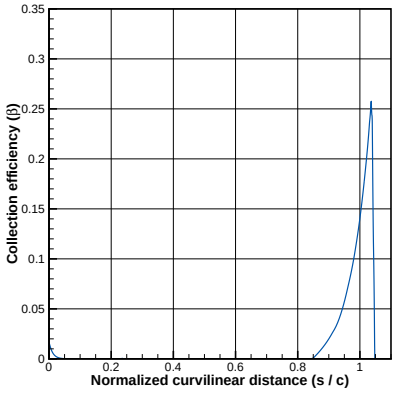
Figure 2.4 shows some results of the droplet solver on the multi-element MDA airfoil : (a) to (c) present the collection efficiency β on the slat, main and flap elements from left to right and (d) presents the trajectory lines computed by interpolating the droplet velocity field. In an Eulerian solver, the collection efficiency is computed by

$$\beta = \alpha \mathbf{u_d} \cdot \mathbf{n_f} \tag{2.1}$$

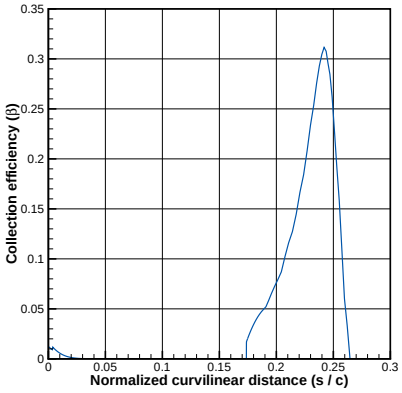
where α is the droplet volume fraction computed by the droplet density equation, $\mathbf{u_d}$ is the droplet velocity vector and $\mathbf{n_f}$ is the facet normal vector on a solid boundary. This collection efficiency is directly converted in water impingement rate to compute the ice accretion rate on the surface.



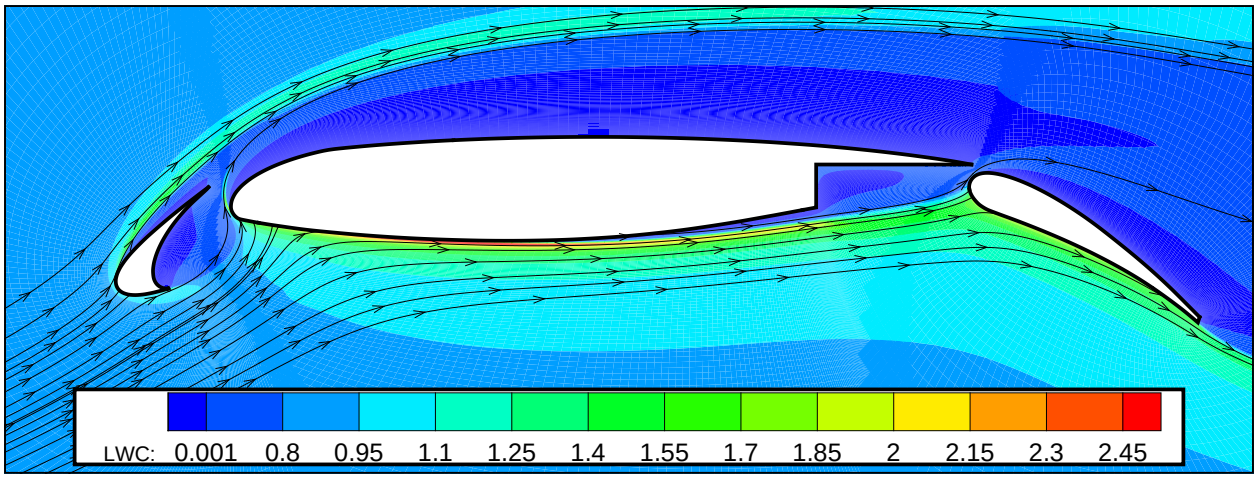
(a) Impingement on the MDA slat element ($s == 0$ at lower trailing edge)



(b) Impingement on the MDA main element ($s == 0$ at lower trailing edge)



(c) Impingement on the MDA flap element ($s == 0$ at lower trailing edge)



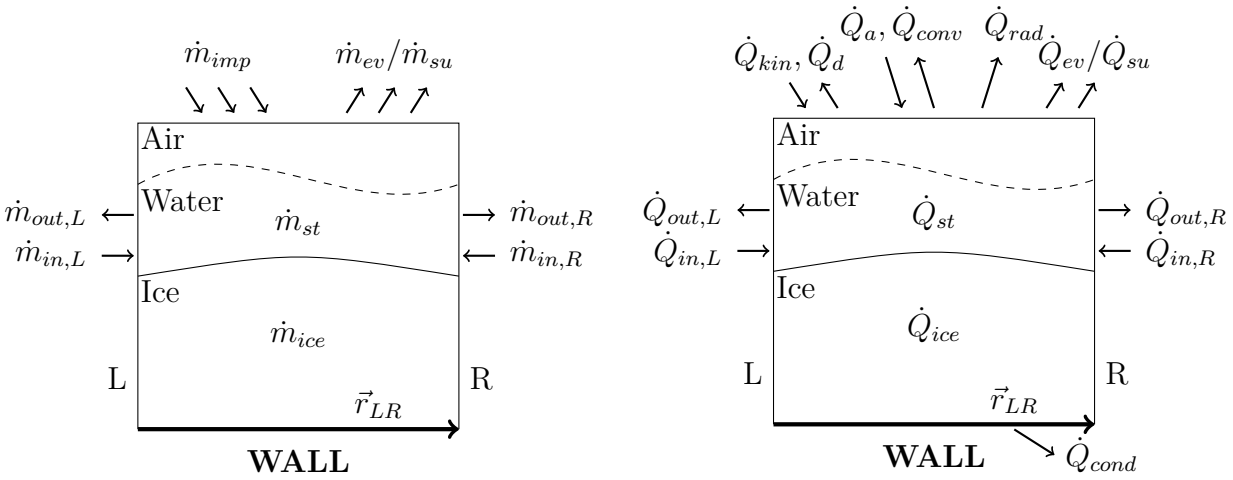
(d) Trajectory lines and LWC (g/m^3) contour around the MDA

Figure 2.4 Droplet trajectory lines and impingement on a multi-element MDA airfoil

2.4 Thermodynamic surface exchanges

Knowing the water impingement rate and the aerodynamic convection heat transfer on the solid surfaces, mass and energy balance equations can be used to obtain the ice accretion mapping on the surface. For these two equations, three unknowns are sought : i) the ice accretion rate ($\dot{m}_{ice}$), ii) the water runback flux ($\dot{m}_{out}$) and iii) the surface temperature (T_s). Figure 2.5 shows the mass and energy balances on a control volume with ice, liquid water and air with runback water exchanges with neighbour control volumes on the left (L) and right (R) sides of the volume. The system can be completed by adding state equations, according to the work of Messinger [59] which is decisive for 2D icing simulations history. Four states are usually defined to complete the thermodynamic balance equations to form the well-known Messinger model.

- No water : The trivial state without water.
- Dry surface : When all the incoming water freezes, implying that $\dot{m}_{out} = 0$.
- Wet surface : With a water film freezing in part only, implying that $T_s = T_{freeze}$.
- Liquid surface : Without ice formation, implying that $\dot{m}_{ice} = 0$.



(a) Mass balance

(b) Energy balance

Figure 2.5 Control volume diagrams for mass and energy balance equations

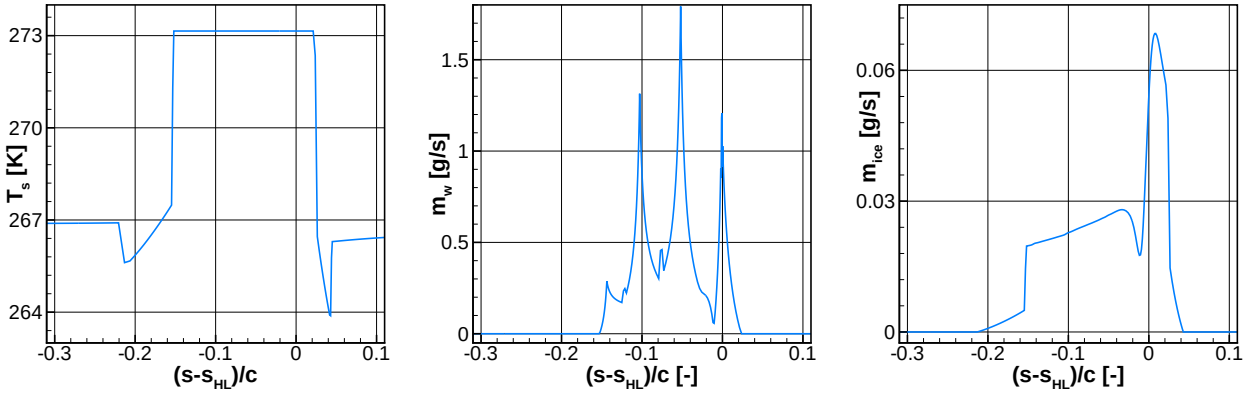
The issue in the classical Messinger model is the dependency to the stagnation point on an airfoil which is used to determine the runback water flow direction. As the equations are solved algebraically, the remaining water at the stagnation point is simply sent in equal proportion to the neighbouring surface facets and from there the lower and upper regions of the airfoil are computed facet by facet.

Different models were created to remove this global dependency and provide a local thermodynamic model possible to extend in 3D. Zhu *et al.* [112] embedded the Messinger model in an iterative loop where the runback water flow direction is defined with the air shear stress and the pressure gradient in the water layer, thus removing the dependency on the stagnation point location. In a tentative to better model the freezing process, Myers *et al.* [63] extended the Messinger model by incorporating the conduction through ice computed with the difference of temperature between the airfoil surface and the ice surface. However, this model requires the knowledge of the stagnation point location and of the airfoil surface temperature, which is usually unknown for cases without anti-icing. It was later modified to remove the stagnation point dependency by determining the runback water direction from the air shear stress, the pressure gradient in the water layer and the gravity forces [63].

In another development direction, a PDE-based model using the air shear stress to determine the runback water direction was presented by Bourgault *et al.* [10] : the Shallow Water Icing Model (SWIM). It is based on lubrication theory and is easily extended in 3D. However, because it models the freezing process in an unsteady way where the ice and water film heights are solved for, the computing cost of the SWIM is high.

2.4.1 Iterative Messinger model

Following the work of Lavoie [54], different thermodynamic models are implemented in NSCODE-ICE : i) the classic Messinger model [59], ii) the Iterative Messinger model [112], iii) the extended Messinger model [63] and iv) the SWIM [10]. However, only the iterative Messinger model is used in the present study as it was found to be the most efficient and to give similar results when compared to the SWIM [54].



(a) Surface temperature ($s == 0$ at high-light point)

(b) Outgoing water runback flux ($s == 0$ at high-light point)

(c) Ice mass accretion rate ($s == 0$ at high-light point)

Figure 2.6 Examples of the thermodynamic model results on a glaze ice case over a NACA0012 airfoil

This model starts by assuming that there is no runback water flux between the surface facets. Iterations are performed to solve the algebraic thermodynamic balance equations on each facet and the runback water flux are updated at each iteration depending on the residuals from the equations. The runback water direction is obtained by following the air shear stress direction on the surface.

Figure 2.6 shows some results obtained with the iterative Messinger model on a NACA0012 airfoil in glaze ice conditions. In Figure 2.6(a), the final surface temperature is shown, with the plateau indicating the glaze ice area. Figure 2.6(b) shows how the runback water flows on the surface, starting naturally from the stagnation region toward the upper and lower regions of the airfoil. Finally, Figure 2.6(c) shows the ice accretion mapping on the surface

which is used in the geometry evolution process to predict the ice shape. The horizontal axis represents the non-dimensional curvilinear distance from the highlight point at the leading edge ($(s - s_{HL})/c$).

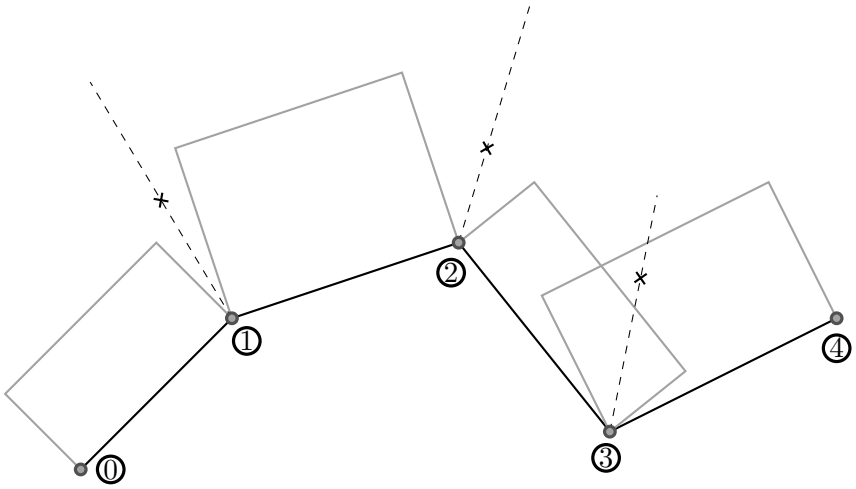
2.5 Geometry evolution

With the ice mass accretion mapping on the surface, the geometry can be updated. This part is critical : i) because the ice shape is the end result of the global simulation, and ii) because it is the source of most failures in an icing software. It must model the physics of geometry deformation adequately, but also be robust enough to handle all particular cases which happen in the discrete world of numerical simulations. However, despite the importance of this module, it receives less attention in the literature. Except in papers dedicated to the subject, a large portion of the literature is not even mentioning the techniques used in the studies to modify the geometry and it is often assumed that the classical algebraic approach is employed in these cases.

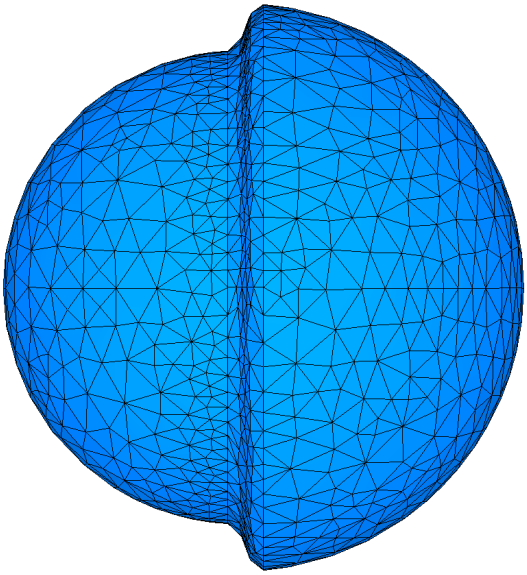
The algebraic method, also called Lagrangian geometry method, consists in moving the vertices of a surface mesh in a defined local direction depending on the amount of ice which needs to be accumulated. In 2D, this direction is often taken as the bisector between neighbouring surface facets [60, 19]. The method in its simplest expression does not conserve the ice mass, as vertices are moved according to the average of ice heights computed on the facet centre from a normal displacement. Figure 2.7(a) shows this process. It is also possible to start the update from the stagnation point, if it is known, and then displace the neighbour vertices toward the trailing edge (TE) with the exact volume of ice [35]. Using iterations also helps to reduce the error on the final volume [79], but might increase the risk of obtaining oscillations in the final solution as it was observed during this study.

These algebraic methods can be used in 2D, but are not suited for 3D modelling. To overcome this issue, Thompson *et al.* [96] presented an algorithm to deform a surface mesh made of triangles for an icing simulation. The facets are directly displaced with a special smoothing method aiming to preserve the volume. The process is performed iteratively with each step being limited to the maximum possible displacement allowing for a valid geometry. Figure 2.7(b) shows the capabilities of this method on a sphere with the right half receiving a uniform amount of ice. However, as the method is explicitly defining the interface, there still exist a risk of obtaining an invalid geometry in concave regions or an inaccurate geometry definition in convex regions if vertices are displaced too far apart.

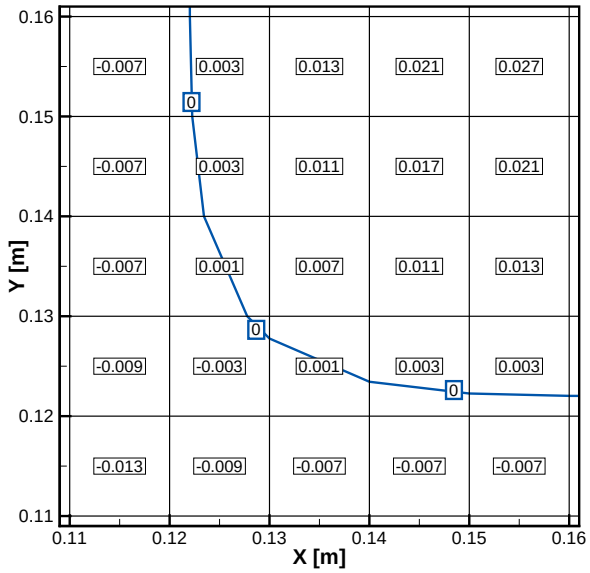
Using a completely different approach, Pena [72] developed a geometry update process using



(a) Algebraic normal geometry update



(b) Iced unstructured sphere



(c) Level-set implicit interface definition

Figure 2.7 Examples of the geometry evolution methods. Figure 2.7(b) adapted from Thompson *et al.* [96]

the level-set equation (Eq. (2.2)), which is an interface transport PDE that captures implicitly the ice interface through a scalar signed distance field (ϕ). Figure 2.7(c) shows an example of how the implicit front is defined.

$$\frac{\partial \phi}{\partial t} + \mathbf{V} \cdot \nabla \phi = 0 \tag{2.2}$$

Also used to treat problems with free surfaces, the level-set equation is applied in icing to

manage the changes of topology that might occur when two ice horns merge together or when ice breakup happens. However, a well-known drawback of this method is the lack of mass conservation, which is critical in ice accretion. Recently, many studies were presented on how to solve the level-set equation for deforming geometries while conserving the mass [4, 81]. Another added complexity when using such an implicit interface in icing reside in the extraction of an explicit geometry used to generate the next body-conforming grid for the next layer computation in the multi-layer process.

The volume-of-fluids method is also used in icing, but on a very small scale to study liquid droplets splashing at impact [1] or freezing on ice-phobic surface coatings [3]. It is similar to the level-set method in the sense that it models an implicit interface, however, the equations are solved to obtain a scalar field directly describing the phase in which a cell is located (fluid or solid or in between). As it allows a cell to be in a middle state, the volume is inherently conserved, which is an advantage compared to the level-set method. However, its main drawback is that it requires a very fine grid around highly curved solids, ideally with Cartesian cells, which increases the computational cost beyond feasibility for aircraft applications.

2.5.1 Iterative algebraic method

Before the current study, an iterative algebraic method was the only available method in NSCODE-ICE to model the iced geometry evolution. It is used in this study as a comparison point. In this method, the geometry is updated in multiple steps, also called sub-layers, instead of in one step, allowing to consider the geometry change in the process, as shown in Figure 2.8. The ice thickness to achieve for a sub-layer l_{sub} is corrected at each iteration to account for the update in curvilinear distance, i.e. for the change in geometry length, yielding Eq. (2.3).

$$h_{ice}^{l_{sub}} = \frac{h_{ice}\ \Delta s}{N_{sub}\ \Delta s^{l_{sub}}} \tag{2.3}$$

Here, h and $h^{l_{sub}}$ represent the total and sub-layer ice thickness at a vertex, Δs and $\Delta s^{l_{sub}}$ are the initial and sub-layer curvilinear distance between vertices and N_{sub} denotes the number of iterations to perform.

Then, the ice is grown by node displacement following the direction of the bisectors (represented by dashed lines on Figure 2.8). This approach tends to limit the errors in mass conservation when $N_{sub} \to \infty$ by accounting for the curved geometry. For the ice accretion

simulations using the algebraic method presented in this work, $N_{sub} = 20$ has been found to provide a sufficiently small error on the ice mass for each sub-layer (error $\approx 0.05\%$).

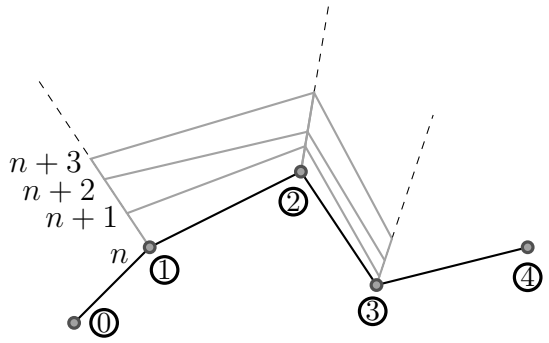


Figure 2.8 Iterative algebraic method example

Note that the icing simulations in this work are performed using a multi-layer approach not to be confused with the sub-layers presented in this section. The layers in the multi-layer approach are used to ensure proper representation of the physics of the ice accretion by updating the airflow, droplets field and thermodynamic balances at each layer while the sub-layers are applied during the ice growth of each layer to limit ice mass loss or gain when using the iterative algebraic method.

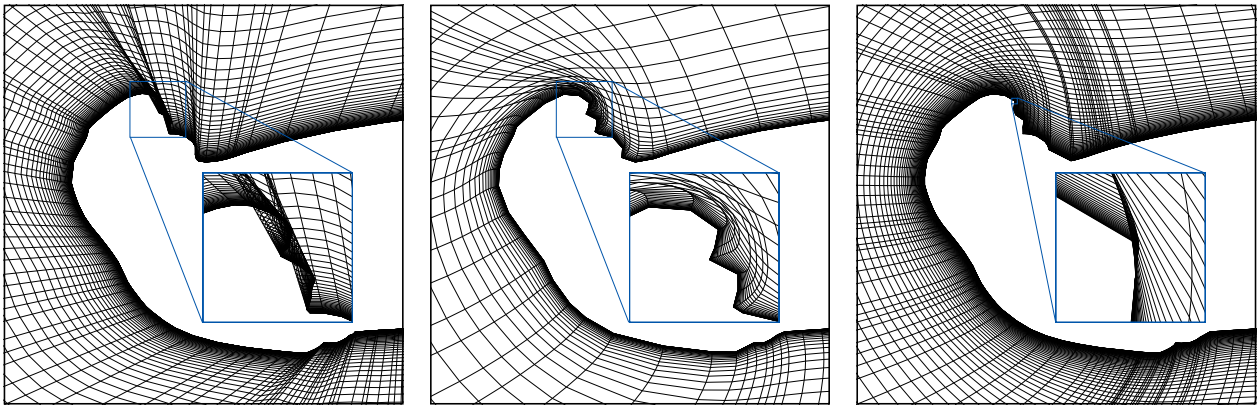
2.6 Volume mesh discretization

To solve any of the previously described physical modules, a spatial discretization of the solved domain is required. The use of a potential method, as in the first generation of icing tools, is advantageous in this aspect because only a surface grid is required. The introduction of the RANS equations in icing requires instead volume grids, which are more complex to obtain with high quality on complex configurations. Even in 2D, meshing the domain around an ice shape with a structured grid is a challenging task [46].

2.6.1 Elliptic mesh generation

In NSCODE-ICE, a structured grid generation tool initially developed by [46] is used to produce high quality grids on airfoils. A Transfinite Interpolation (TFI) algorithm is used to generate an initial algebraic grid which is then smoothed with an elliptic PDE system [44]. Different functions are added to the smoother to control the grid spacing, the orthogonality and the skewness of the cells. However, the use of a TFI algorithm to generate the initial algebraic mesh is limiting the capacities of the mesh generation, although the strong elliptic smoother is usually able to yield a valid grid even with highly deformed algebraic grids such as

can be obtained in icing. An example of how the elliptic grid generation performs on horn ice shapes is shown in Figure 2.9(a) for the invalid TFI mesh and Figure 2.9(b) for the smoothed grid which is valid, but of low quality. Figures 2.10(a) and 2.10(b) show this low quality with respect to the orthogonality and skewness criteria visualized with a contour where the dark green areas are of low quality (near 0 for the orthogonality and near 0.5 for the skewness). Figure 2.9(c) shows another grid generated with the elliptic method on the same geometry where fewer iterations were used in the smoother, resulting in an invalid mesh. This shows that for complex geometries, the elliptic smoother reaches a limit in robustness and depends on user parameters, which is an issue in a fully automated multi-layer simulation.

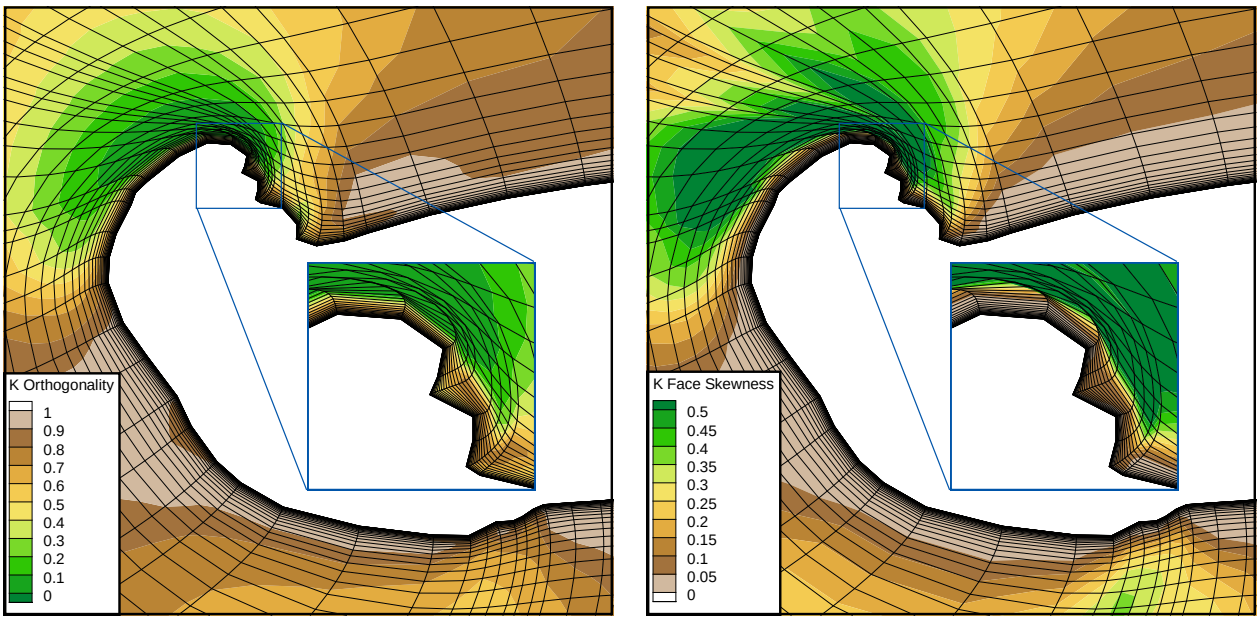


(a) TFI algebraic grid (b) Elliptic smoothed valid grid (c) Elliptic smoothed invalid grid

Figure 2.9 Close view of a coarse grid behind a ice horn for different generation techniques

2.6.2 Overset grid method

As seen in Figure 2.2(c), NSCODE-ICE is compatible with overset grids. The cells not shown are considered interpolated cells and receive data from donor cells through an interpolation step. The donor cells at the edge of the computed zone act as ghost cells for any other boundary. The overset grid method is very useful when dealing with multi-element airfoils in structured grids as well as unstructured grids [48]. It is also useful for unsteady simulations or internal simulations with airfoils, such as wind tunnel simulations. The technique requires : i) to find which cells are computed and which cells are interpolated depending on a wall distance criterion, and ii) to compute weights from donor cells for the interpolations. Among the different interpolation methods that are implemented in the software, the bilinear method gives the most accurate results (2^{nd}-order) with a relatively low cost [56].



(a) Orthogonality criterion

(b) Skewness criterion

Figure 2.10 Quality of a grid generated with the elliptic method on a ice horn geometry

2.7 Strategies for robustness

In the scope of this study, a more robust methodology to perform multi-layer icing is sought. Some strategies to achieve this goal are presented here as they are not directly related to the geometry evolution methods presented in the following chapters. As mentioned, a smooth geometry representation obtained from a CAD should be the basis for any CFD tool [26].

2.7.1 Continuous geometry support

When discussing about smooth geometries from CAD, NURBS is often the first name that comes to mind, but simpler non-rational B-splines are also used. A B-spline is a continuous curve which is defined by N_{CP} control points, an approximation degree (p) and a knot sequence [27], as shown in Figure 2.11. The curve itself is constructed piecewise from B-spline basis functions that are built recursively from the degree 0 to the degree p and that affect the curve only in a number of surrounding knot spans. Depending on p and on the multiplicity of the knots (N_k), the curve is locally differentiable $p - N_k$ times. Additionally, a NURBS contains rational non-uniform weights which multiply the basis functions and allow a better control on the curve and to conserve the curve in projection and affine operations whereas B-splines are conserved only for the latter [73].

However, these curves come with a price in numerical complexity which means it is not real-

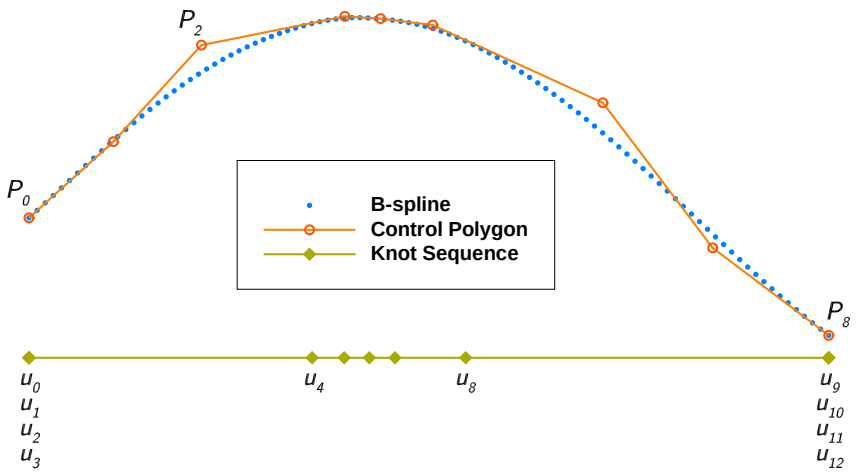


Figure 2.11 A cubic B-spline built with nine control points and thirteen knots. Figure adapted from Bourgault-Côté *et al.* [16]

istic to develop an in-house tool for this aspect. Furthermore, commercial CAD tools which handle NURBS usually do not give access to the numerical definition of the curves, which would be of help when approximating the ice shapes and adding it to the previous geometry. OpenCascade [23] is one example of open source CAD software where the mathematical representation of the curves is accessible, but it represents a massive amount of work to bring these equations in a usable format for aerodynamic simulations.

To solve this issue, Haimes and Dannenhoffer [40] developed Engineering Sketch Pad (ESP), a software built on OpenCascade and dedicated to handling aircraft geometries for CFD by providing a low set of specialized functions into a geometry kernel named EGADS. It allows the generation and modification of any shapes required to build a geometry representation of an aircraft or an airfoil. Furthermore, EGADS can be integrated directly into any CFD software to read geometry files, modify parts of a geometry and directly use the geometry for special operations, such as surface discretization or local surface normal vector evaluation. One downside to EGADS is the limitation to B-spline curves instead of NURBS, but it offers enough modelling capacities for 2D geometries and would limit a 3D geometry generation only if projections were required [73].

2.7.2 B-Spline approximation method

To ensure a better approximation of the B-splines on ice shapes, an algorithm was developed with Dr. Docampo-Sànchez of MIT based on a heuristic approach that combines a least square minimization [73] together with an error controlled iteration [15]. The goal is to look

for an approximation that minimizes the error (distance to the data points) while employing a minimum number of control points. Its pseudo-code can be seen in Algorithm 1. Starting with a minimum number of control points, the knot span with the greatest local error (ϵ_{loc}) is looked for and a new knot is inserted in it before finding the control points using a linear least square solver. If the new approximation improves the curve error more than the minimal step tolerance ($-\Delta\epsilon > \epsilon_\Delta$), meaning that the addition of the knot affects significantly the curve, the knot sequence is updated. Otherwise, the knot is rejected and a new knot is inserted at the next "high-error" span. When no new knots overcome the step control, the algorithm takes the "best so far", tags it, iterates and when a valid step is found, attempts to remove any potential redundant knots. This heuristic process has been devised to bypass spans featuring a high error, but affecting less the curve than other knot spans. The routine stops when either (i) the global error hits the approximation tolerance target (ϵ_{curve}), (ii) it reaches the maximum number of iterations or (iii) no new knots can improve the current approximation.

Algorithm 1 B-spline fitting

```
1:  while ϵ ≥ ϵ_curve do
2:      for i do
3:          if ϵ_loc,i = max(ϵ_loc) then
4:              Split knot span at middle
5:              if −Δϵ ≤ ϵ_Δ then
6:                  Reject insertion
7:              end if
8:          end if
9:      end for
10:     if No insertion then
11:         Take "Best so far"
12:     end if
13:     Removal of redundant knots
14: end while
```

Figure 2.12 shows the performance of the algorithm on two sets of data points : the NACA0012 airfoil for which the data points are generated using the NACA airfoil equation and an ice shape obtained on a manufactured airfoil for which the data points were extracted in a simulation using the algebraic method. Note that the Y-axis and the X-axis are stretched in Figure 2.12(a) and Figure 2.12(b) respectively, with the legend in Figure 2.12(a) applying to

(a) Closed airfoil.

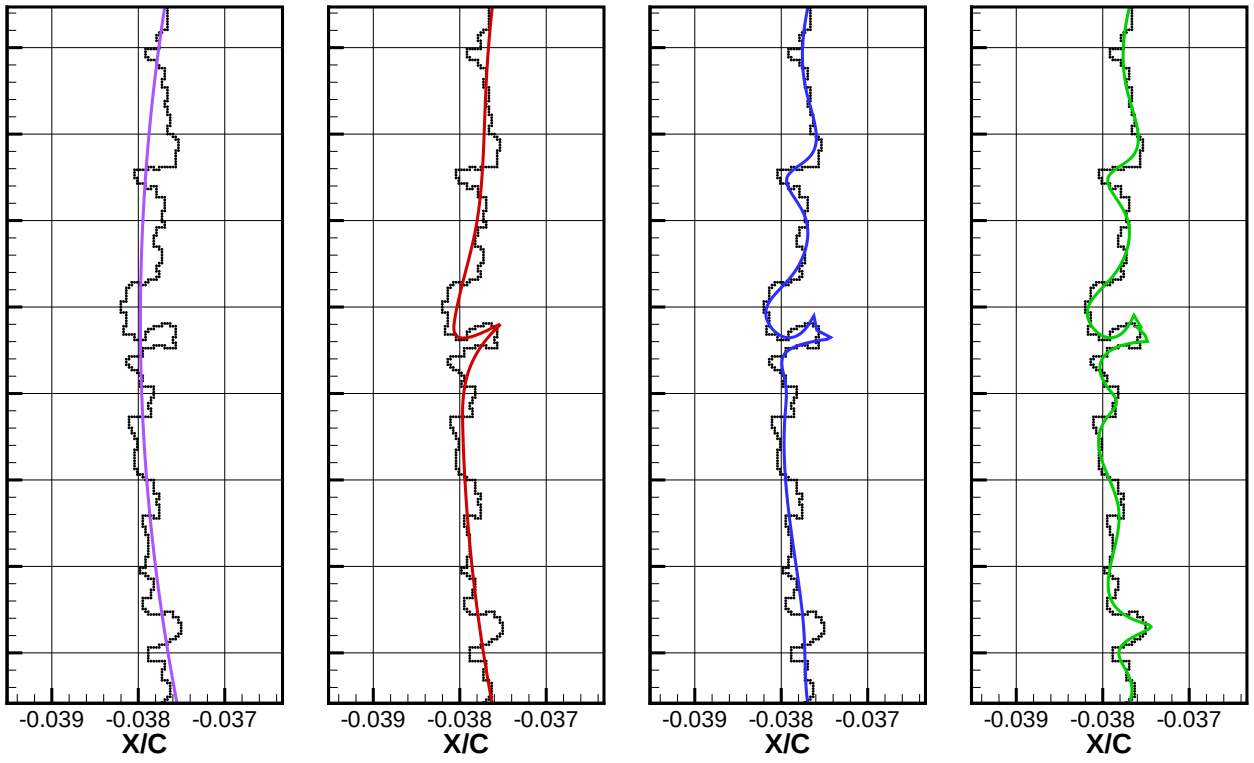
(b) Extracted ice shape.

Figure 2.12 Cubic B-spline reconstructions for two different data sets using (a) 27 and (b) 45 control points. Figures adapted from Bourgault-Côté *et al.* [16]

both.

Effect of the approximation tolerance

The control offered by the curve approximation tolerance (ϵ_{curve}) in Algorithm 1 allows to capture different levels of detail in a set of data. Figure 2.13 shows four different curve approximations obtained with the same data set with different curve tolerances. As the tolerance becomes stricter, the curves are closer to the data points, more control points are generated and smaller features are captured. In fact, from Figures 2.13(a) to 2.13(d), the curves stay similar globally, but a localized area varies in each Figure compared to the precedent one where local data points feature are being capturing with stricter tolerances. This highlights the error-controlled part of Algorithm 1 in which the knot span improving the most the curve is selected to insert control points. In a standard algorithm where the knot distribution is uniform, the curve would be modified everywhere at each iteration.

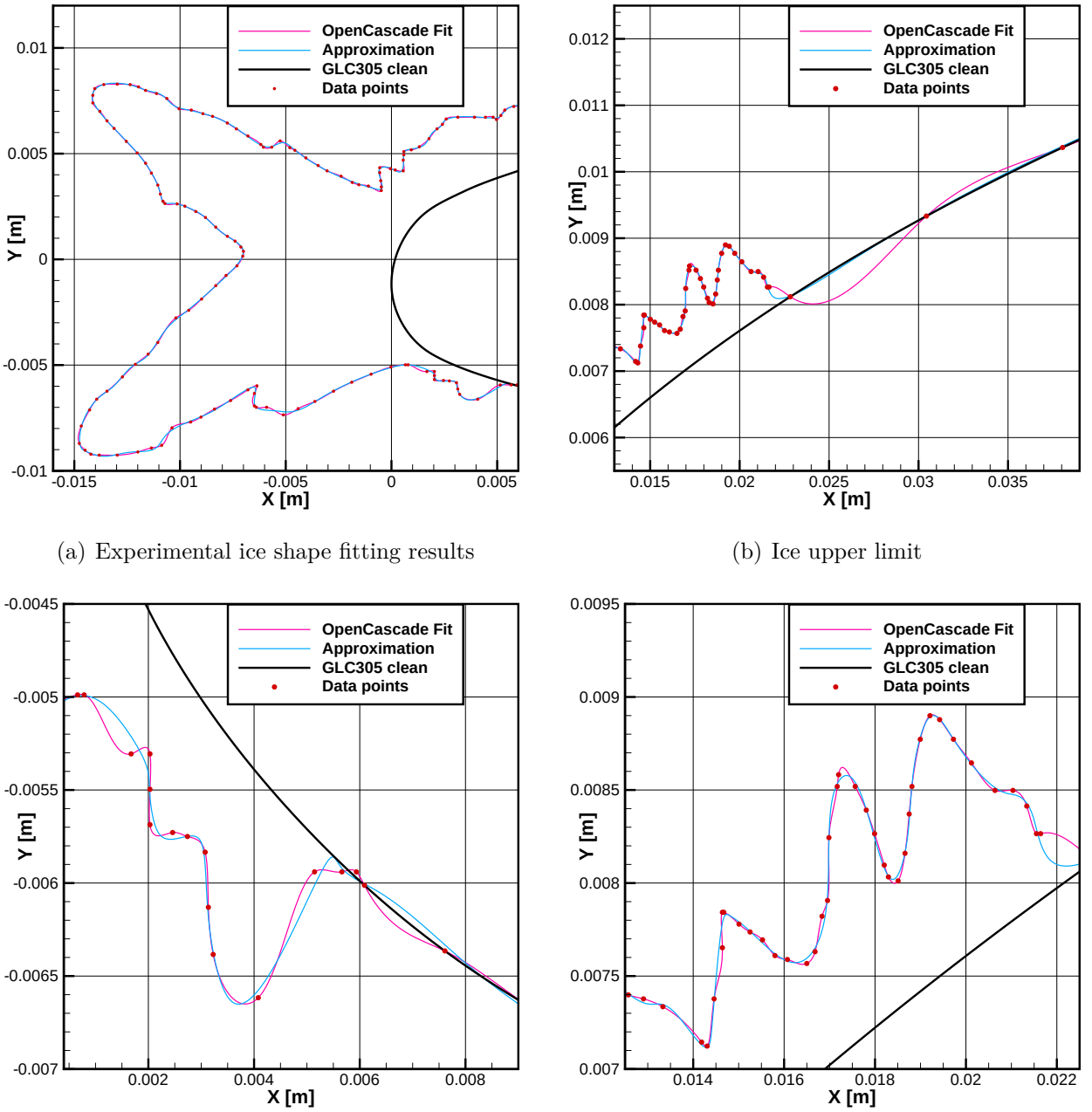


(a) ϵ_{curve} = 1.0e−4, $N_{CP} = 7$ (b) ϵ_{curve} = 3.5e−5, $N_{CP} = 14$ (c) ϵ_{curve} = 2.75e−5, $N_{CP} = 51$ (d) ϵ_{curve} = 2.0e−5, $N_{CP} = 107$

Figure 2.13 Effect of the approximation tolerance on the B-spline curve

Comparison with fitting method

To further assess the capacities of the approximation method, it is compared in Figure 2.14 to a fitting method from OpenCascade. The main difference between a fit and an approximation



(a) Experimental ice shape fitting results

(b) Ice upper limit

(c) Ice lower limit

(d) Ice feathers

Figure 2.14 Comparison between the approximation method of Algorithm 1 to the fitting method from OpenCascade on an experimental ice shape (GLC305 airfoil)

is that the former tries to pass by the data points whereas the latter tries to pass through the data points to respect a tolerance criterion. The comparison is performed on an experimental ice shape obtained on a GLC305 airfoil [100]. Globally, both methods yield a similar result (Figure 2.14(a)), but the fitting method requires 274 control points while the approximation method uses only 114. Furthermore, the approximation method results in a smoother junction

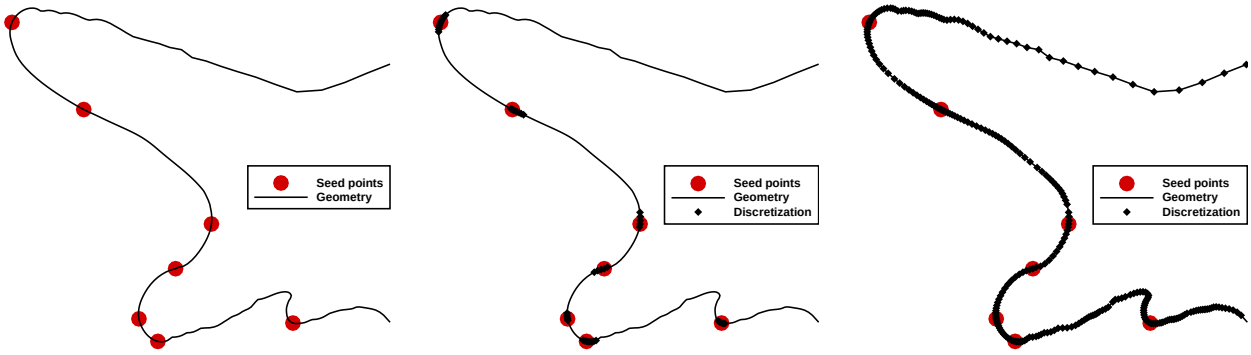
with the portion of the data corresponding to the initial GLC305 airfoil, as seen in Figures 2.14(b) and 2.14(c). Although the fitted curve pass closer to the data points, e.g. on the rime feathers seen in Figure 2.14(d), it results in a more oscillating curve that might affect the aerodynamic flow simulations, which are known to be sensitive to the geometry curvature variations. Finally, the distribution of the control points in the approximated curve is centered around the iced portion of the data, with only 17 control points (15% of N_{CP}) in the clean portion compared to 90 (33% of N_{CP}) for the fitted curve. On a small 2D curve, this increase in memory to represent the curve is negligible, however, for 3D surfaces on a complete aircraft configuration, it could represent an issue.

2.7.3 Curvature-based surface discretization

One-dimensional (1D) discretization methods integrated in NSGRID provide different patterns to discretize a smooth curve in order to define the boundaries of a grid being generated. Among those methods, a curvature adaptive method is implemented based on the tangent vector variation and on the signed curvature. The discretization is performed in a similar way as an advancing-front unstructured mesh generation algorithm and the main steps are illustrated in Figure 2.15. The curve is first uniformly discretized very finely and seed points are found based on different criteria (Figure 2.15(a)): i) a maximum of the first derivative norm (rate of change of the tangent vector), ii) a sharp angle between two consecutive tangent vectors or iii) a user defined position on the curve. Some seed points are removed when they are too close to another seed point featuring a higher curvature, which happens regularly in ice shapes. A new vertex i is placed at a distance $\Delta s^i = \gamma \Delta s^{(i-1)}$ where γ is the growth ratio which is kept constant on all the geometry to enhance the global quality, but also the junctions between two advancing fronts (Figure 2.15(b)). The parameter Δs^1 at the seed points depends on user inputs and curvature sign to coarsen concave regions and refine convex regions as recommended [24]. From the different seed points, the curve is discretized by adding vertices in an ascending order depending on the distance between vertices at the fronts ($\Delta s^{(i-1)}$) to obtain an acceptable ratio at the junction between two advancing fronts (Figure 2.15(c)). One advantage of this method is that small ice features are captured in the surface mesh.

2.7.4 Hyperbolic mesh generation

As mentioned in §2.6, combining algebraic grid generation and elliptic grid smoothing allows to generate high quality grids on simple geometries, but limits the automation when dealing with complex geometries such as ice horns. To solve this issue, a hyperbolic mesh generation



(a) Find seed points (b) Start fronts (c) Fronts advance

Figure 2.15 Main steps of the curvature adaptive 1D discretization method

algorithm [25] is added to the various tools used in this study to improve the quality of the grids generated during the multi-layer icing simulations. The previous elliptic method can result in an invalid mesh with negative volumes even when the underlying B-spline geometry is valid due to the constraint to link the wall with a pre-discretized farfield. However, this constraint is removed in the hyperbolic method. Furthermore, the hyperbolic mesh generation is based on the resolution of a PDE-based system of equations and is thus more robust than the coupling between a TFI algorithm and an elliptic smoother. The mathematical details can be found in Appendix A, with a brief introduction given below.

For a planar 2D grid, the hyperbolic field grid generation equations are obtained from two constraints [51, 24]:

$$\mathbf{r}_\xi \cdot \mathbf{r}_\eta = 0 \tag{2.4a}$$

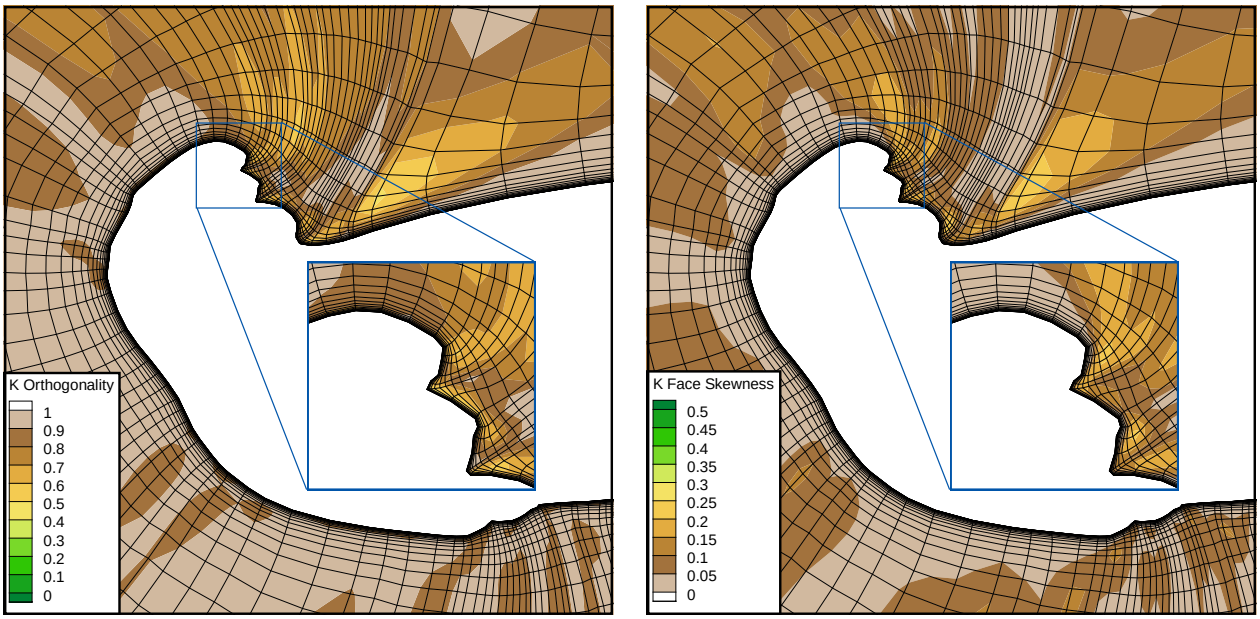
$$\hat{\mathbf{n}} \cdot (\mathbf{r}_\xi \times \mathbf{r}_\eta) = \Delta V \tag{2.4b}$$

where $\mathbf{r}_\xi$ and $\mathbf{r}_\eta$ are the vector derivatives in the ξ and η directions of the grid at a vertex and $\hat{\mathbf{n}}$ is the plane normal. In this context, ΔV is the local cell volume which is obtained around a vertex when adding a grid layer in the η direction. The constraints are linearized and the time-like marching is performed in the η direction [24]. A block-matrix system made of three diagonals is then obtained and can be solved with an iterative method.

To further increase the robustness of the method and add stability to the equations, different smoothing methods are added [25] and implicitness is introduced along with artificial dissipation [51]. It allows to handle highly concave or convex regions robustly and produce high quality grids in these regions which is always a challenge in grid generation. Furthermore,

an almost circular farfield is obtained naturally, which corresponds to the desired global grid shape for an external aerodynamic simulation. The details regarding the numerical resolution, the smoothing mechanisms and the choices of parameters are discussed in Appendix A. These smoothing parameters are set to values suggested in the literature [25] to yield high-quality grids for most of the icing cases studied so far, including the cases seen in this study. This represents an important feature in the development of an automated ice accretion software.

Figures 2.16(a) and 2.16(b) show the grid obtained with the hyperbolic method on the same ice horn geometry shown previously in §2.6 with the contour representing the orthogonality (the closer to 1 the better) and skewness (the closer to 0 the better) grid quality metrics, respectively. It shows the high quality grids which the hyperbolic mesh generation method can produce when compared to the elliptic grid quality metrics shown in Figures 2.10(a) and 2.10(b). Note that these grids are generated in a coarse manner for display purpose, which explains the high expansion ratio behind the ice horn.



(a) Orthogonality criterion

(b) Skewness criterion

Figure 2.16 Quality of a grid generated with the hyperbolic method on a ice horn geometry

2.8 Conclusion

In this chapter, the multi-layer icing process was detailed from a global perspective. In particular, the hypotheses made to simplify the numerical representation of the process are justified. Each module composing a multi-layer icing simulation is described in regard to

historical evolution of the technology and the related models implemented into NSCODE-ICE that are used in the present study are described with more details. In each cases, solution examples are given to highlight the capacity of NSCODE-ICE to model the physical phenomena involved in aircraft icing.

In the last section of this chapter, different strategies to enhance the multi-layer global process are described . The choice of B-splines from EGADS [40] as a smooth geometry support for NSCODE-ICE is justified and the performance of the approximation algorithm developed specifically for ice shapes is demonstrated. The methods implemented to improve the curves and volume grids are described. The use of a curvature adaptive discretization technique on the geometries allow to better capture convex or concave regions. Finally, the hyperbolic volume mesh generation algorithm from Chan and Steger [25] is implemented and allows to produce automatically higher quality volume grids than the elliptic generation methodology previously used.

CHAPTER 3 LEVEL-SET ICE ACCRETION

In the multi-layer icing process described in Chapter 2, one of the modules identified as having high potential for improvement is the geometry evolution. In order to propose a robust approach that models adequately the physical behaviour of freezing water on an airfoil, the developments of a level-set method for ice accretion are detailed in this chapter. The impact of the numerical developments on the external field velocity and the spatial accuracy are presented. In particular, strategies to obtain mass conservation are presented and results are compared to the algebraic method and to experimental data.[1]

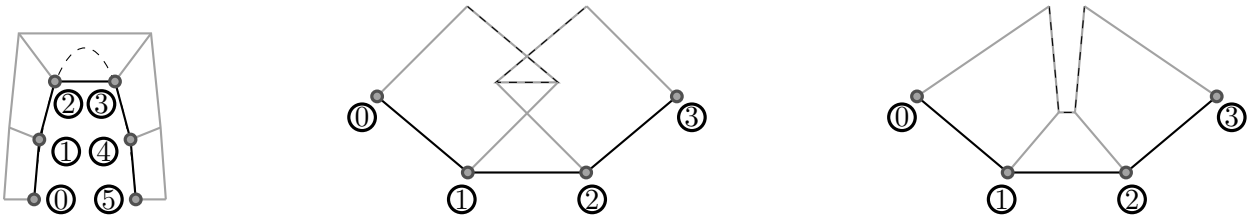
3.1 Context

Interface-tracking methods such as the algebraic method traditionally used in icing simulation software deform a geometry by explicitly moving its constituting points. In the case of icing, the displacement direction is usually defined as normal to the surface mesh by using the bisectors between surface facets, as presented briefly in §2.5. However, such explicit displacement presents some issues when deforming geometries that are then used in simulations. The different problems can be classified according to the local signed curvature of a geometry: convex or concave.

In convex regions, the new points are more distant than the initial points are, compromising the accuracy with which the geometry features are captured. If the displacement in these regions is too high, the surface mesh points need to be redistributed before generating volume meshes to ensure the geometry is correctly simulated, which is difficult when there is no underlying smooth geometry. Figure 3.1(a) illustrates this issue where the upper facet doubles in length after the deformation and the underlying curve between nodes 2 and 3 is ignored in the surface mesh displacement even though it features a highly convex part. Dividing the oversized segments in two once it reaches a tolerance is a viable solution [84], but comes with an increase in algorithm complexities as it must then be treated in an unstructured manner.

In concave regions, the node displacement can cause the most critical robustness issue in geometry deformation, that is the production of invalid elements when self-intersections are produced in the explicit interface [84]. Figure 3.1(b) shows this phenomenon with the first segment intersecting the last segment in the displaced layer shown with a dashed line. Different methods exist to detect these self-intersections and remove the invalid segments, but

[1] Parts of this chapter are based on references [13, 15, 16].



(a) Convex region over coarsening

(b) Concave region self-intersecting

(c) Concave region almost self-intersecting

Figure 3.1 Examples of issues caused by an explicit surface mesh deformation in convex and concave regions

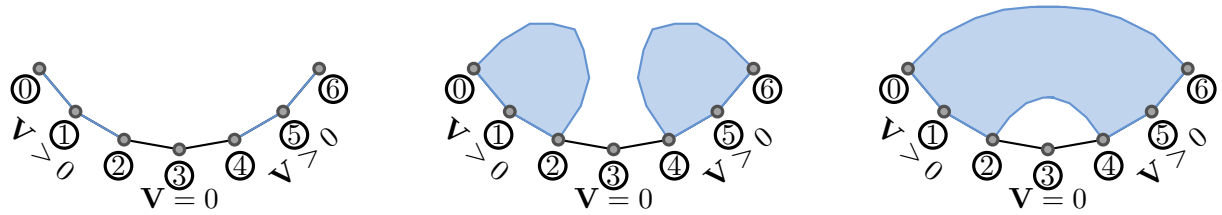
the accreted volume associated with these segments is then lost, which is another problem in icing. However, even with the appropriate methodology, the self-intersection does not really need to happen to cause an issue. A case is shown in Figure 3.1(c) where the two parts of the interface surrounding the concave region are displaced without causing intersections, but impeding the generation of a new volume grid due to the small distance between the non-consecutive segments.

Different approaches are presented in the literature to reduce the risk of obtaining these issues, such as dividing the displacement process in iterations. Limiting the displacement distance and smoothing between iterations have shown good results [96]. Other methods also redistribute the surface mesh points between the iterations to ensure a smoother geometry evolution [79]. However, these techniques are used to compensate problems inherent to the equations themselves. Better equations to model the ice evolution are thus required. Furthermore, in the development of new icing software based on PDE models, a PDE-based geometry evolution method is sought to benefit from a generalization and from already implemented solving tools.

3.1.1 Implicit interface-capturing

On contrary to the explicit methods, interface-capturing methods model an interface implicitly by solving a scalar state of an element, such as the distance to the interface [68]. These methods inherently have the property to merge two topological advancing fronts that encounter each other as they are represented by scalars in a domain. It represents a solution to the problems of self-intersections produced by explicit methods in concave regions. This can be visualized on Figure 3.2 where two moving fronts surrounding a fixed front section merge together and separate the fixed region from the rest of the domain. The volume-of-fluid is an example of interface-capturing method where the interaction between fluids is solved on very

fine Cartesian grids [81], but it is too expensive for aircraft icing applications as mentioned in §2.5. The level-set method is another example and it was used in icing to model the freezing process of water droplets [101].



(a) Fixed concave region

(b) Neighbour fronts almost in contact

(c) Merged fronts with hole

Figure 3.2 Front merging process of the level-set equation

The level-set equation (Eq. (2.2)) introduced by Osher and Sethian [69] is a simplification of the Hamilton-Jacobi equation applied to dynamic implicit interfaces. It is an Eulerian formulation of the evolution of a scalar variable (ϕ) which is convected depending on an external velocity field. A signed distance function is chosen to represent ϕ as it features interesting properties, such as a unit gradient norm, beneficial to the level-set simulation [68]. The interface is then found at the level $\phi = 0$. In the literature, this method is well known in the field of computer vision and is more often solved on fine Cartesian grids than on curvilinear grids. It also appears that the solution is highly sensitive to the spatial and temporal schemes accuracy [68] and that it requires a proper re-initialization of the solution if the interface is displaced too much, as it degrades the signed function ϕ in the rest of the domain.

In aircraft icing simulations, the level-set equation was introduced recently by Pena [72] in the geometry evolution module of NSMB, a structured RANS solver. The velocity field driving the convection of the level-set interface was extrapolated from the ice accretion velocity at the wall (Eq. (3.9)). First order discretization schemes were deemed sufficient [72], as the scale of the simulation is higher than in computer vision where fourth- or even fifth-order schemes might be required to capture the image details accurately [68]. The results showed a great potential for icing simulations, with an example of two fronts merging easily together and leaving a hole in the middle.

3.2 Level-set equation

According to Osher and Fedkiw [68], the level-set equation (Eq. (2.2)) is a hyperbolic convective equation which is best solved using upwind schemes instead of central schemes as they

are stable by themselves without requiring the addition of artificial dissipation. A simple first order scheme was found to be insufficient to perform image reconstruction with adequate accuracy on Cartesian grids. This matter of fact led to the introduction of essentially non-oscillatory (ENO) schemes to solve the equation with a third order accurate scheme. Further improvements were then proposed to use a weighted version of the ENO scheme (WENO) which allowed to use a fifth order accurate scheme.

In the context of this work, a finite volume solver for the level-set equation is desired to be fully compatible with the other components of the software. This constraint removes the possibility to use the previously mentioned schemes as they are developed for a finite difference method. Frolkovič *et al.* [37] presented a finite volume semi-implicit solver for the level-set equation using the conservative form of the equation as seen in Eq. (3.1) where t is the physical icing time and $\mathbf{V}$ is the interface velocity field.

$$\frac{\partial \phi}{\partial t} + \nabla \cdot (\phi \mathbf{V}) - \phi \nabla \cdot \mathbf{V} = 0 \tag{3.1}$$

The variable ϕ is kept constant in the term containing the divergence of the interface velocity ($\mathbf{V}$). The equation is then integrated in the volume with the flux-divergence theorem applied to obtain the integral form (Eq. (3.2)) of the level-set equation:

$$\begin{aligned}
\iiint_{\Omega} \frac{\partial \phi}{\partial t} \mathrm{d}\Omega + \iiint_{\Omega} \nabla \cdot (\phi \mathbf{V}) \, \mathrm{d}\Omega - \iiint_{\Omega} \phi \nabla \cdot \mathbf{V} \mathrm{d}\Omega = 0 \\
\iiint_{\Omega} \frac{\partial \phi}{\partial t} \mathrm{d}\Omega + \iiint_{\Omega} \nabla \cdot (\phi \mathbf{V}) \, \mathrm{d}\Omega - \phi \iiint_{\Omega} \nabla \cdot \mathbf{V} \mathrm{d}\Omega = 0 \\
\Omega \frac{\partial \phi}{\partial t} + \oiint_{\partial \Omega} \phi \mathbf{V} \partial \Omega - \phi \oiint_{\partial \Omega} \mathbf{V} \partial \Omega = 0
\end{aligned} \tag{3.2}$$

The equation is discretized into Eq. (3.3) and then reorganized following the suggestions from Frolkovič *et al.* [37] to obtain a discrete equation. A fully implicit scheme is then applied to solve the equation, as detailed in §3.2.2.

$$\Omega\frac{\partial\phi}{\partial t}+\sum_{f=0}^{N_f}\left(\phi\mathbf{V}\right)_f\cdot\mathbf{n_f}\Delta S_f-\phi\sum_{f=0}^{N_f}\mathbf{V}_f\cdot\mathbf{n_f}\Delta S_f=0$$

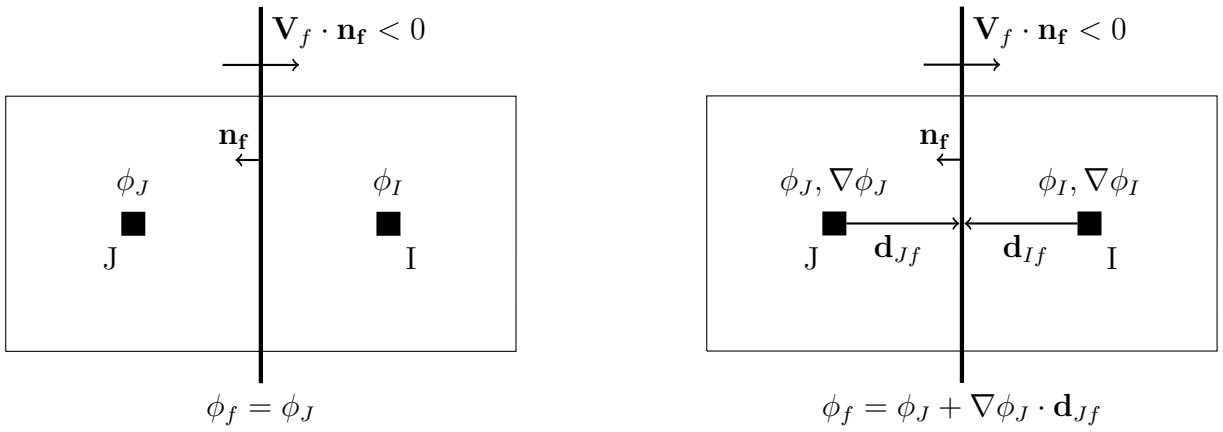
$$\Omega\frac{\partial\phi}{\partial t}+\sum_{f=0}^{N_f}\left[\left(\phi\mathbf{V}\right)_f-\phi_I\mathbf{V}_f\right]\cdot\mathbf{n_f}\Delta S_f=0 \tag{3.3}$$

$$\Omega\frac{\partial\phi}{\partial t}+\sum_{f=0}^{N_f}\left[\left(\phi_f-\phi_I\right)\mathbf{V}_f\cdot\mathbf{n_f}\Delta S_f\right]=0$$

In Eq. (3.3), the subscripts I and f represent respectively a cell-centre value and a facet-centre value and N_f is the number of facets of a cell, which is always 4 in 2D structured grids. The spatial and temporal discretization schemes are then applied to this discrete equation. Note that in the previous equations, Ω and $\partial\Omega$ represent the control volume and its boundaries in a widespread notation that must not be confused with the global volume swept by the interface which is defined in §3.5.

3.2.1 Flux computations

As mentioned, an upwind strategy is required to properly solve the level-set equation in an external velocity field. In this case, first- and second-order upwind schemes are implemented. The upwinding depends on the sign of the dot product $\mathbf{V}_f\cdot\mathbf{n_f}\Delta S_f$ from Eq. (3.3), as seen on Figure 3.3. The value on the side of the facet from which the apparent wind is coming is directly used as ϕ_f in the discretization.



(a) First-order upwind scheme

(b) Second-order upwind scheme

Figure 3.3 Flux computation upwind schemes

The implementation is inspired by unstructured flux computations and the second-order is achieved with a facet-based linear reconstruction of the solution at the facets using the Barth and Jespersen limiter [6] on the gradients. The gradients are computed with a weighted least square algorithm using an extended stencil. A correction is added to avoid odd-even decoupling [8], which was found necessary with the Cartesian grids used in the mesh refinement analysis presented in §3.6. This kind of correction is usually associated with viscous fluxes, which do not exist in the level-set equation. However it is in fact required when gradients are computed cell-centre and then averaged at the facets. In the icing level-set process, the external field velocity ($\mathbf{V}$) is proportional to the interface normal $\mathbf{n}_\phi$ which is the normalized gradient (Eq. (3.4)) and a simple average at the facets of this velocity is therefore insufficient.

$$\mathbf{n}_\phi = \frac{\nabla\phi}{|\nabla\phi|} \tag{3.4}$$

3.2.2 Time integration

The level-set equation must be solved in an unsteady manner and thus requires a time accurate scheme for the temporal discretization. In this work, an implicit second-order backward scheme is chosen [8], although the first temporal integration is performed with a first-order backward scheme to provide an initial previous solution. A dual time-stepping strategy is taken from the flow solver to provide stability and to accelerate the resolution. The temporal derivative in Eq. (3.3) is discretized accordingly and the right-hand side is linearized to include the Jacobian in the left-hand side.

The dual time-stepping strategy allows to use local time-stepping in the field in the pseudo-time iterations and the local time step is defined as follows :

$$\Delta t_{local} = \frac{\sigma \Delta A}{|\lambda_{max}|} \tag{3.5}$$

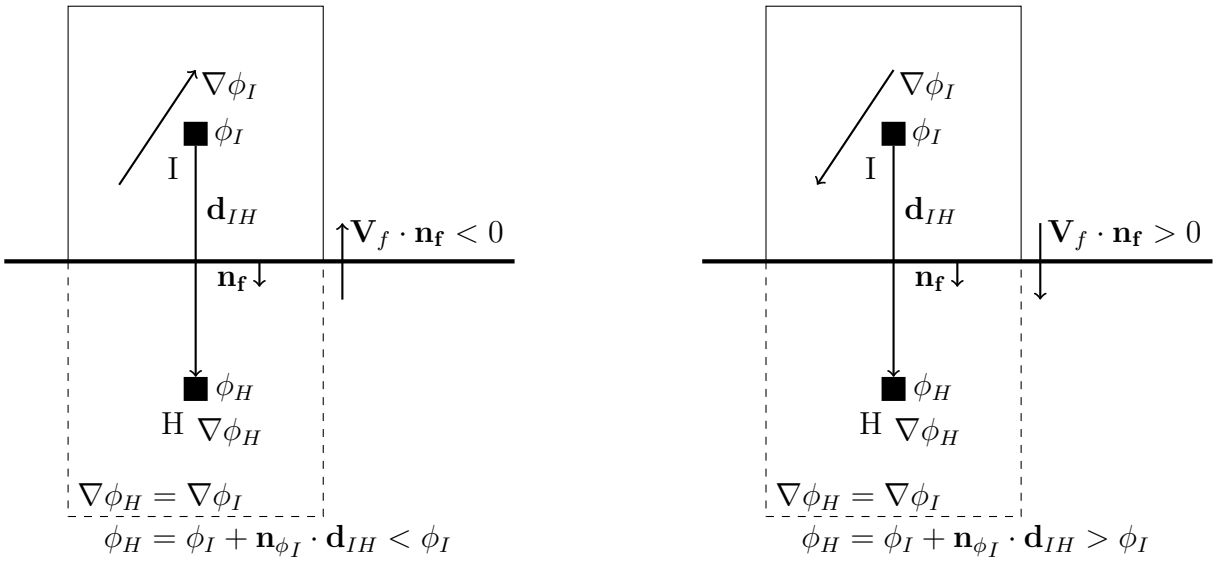
where Δt is the time step, σ is the Courant-Friedrichs-Lewy (CFL) number and $\lambda_{f,max}$ corresponds to the maximal facet eigenvalue of the cell. The eigenvalue is computed by $\lambda_f = \mathbf{V}_f \cdot \mathbf{n_f} \Delta S_f$. Numerical experiments have shown a better convergence when i) the velocity is considered aligned with the facet normal and ii) the velocity at the cell centre is directly used instead of the value at the facet. The eigenvalue equation is then reduced to $\lambda_f = |\mathbf{V}_I| \, \Delta S_f$ for the local time step computation.

A global sparse matrix is constructed for each block of the grid and the system is solved iteratively with a preconditioned BiCGStab algorithm [85]. The physical boundary conditions are treated implicitly in the left-hand side to improve the convergence rate, however, doing so for the connection boundaries did not offer any visible gain.

3.2.3 Boundary conditions

The level-set equation defines implicitly the interface between the air and the ice. As such, imposing a Dirichlet condition would impede the evolution of the interface and only a Neumann condition seems possible. However, the gradient of the level-set is also used in the external velocity field to obtain a normal evolution, therefore it must be treated with care. Considering that ϕ represents a signed distance function and that such function presents a unitary gradient, it indicates that a proper condition would be to impose a unit gradient at the boundaries while conserving the actual gradient orientation. This is obtained through an extrapolation of ϕ at the physical boundaries by using the normalized gradient of the level-set ($\mathbf{n}_\phi$).

The ghost cells are filled by using the same principle as the facet linear reconstruction presented earlier, but with a distance vector from the interior cell centre to the ghost cell centre, as seen in Figure 3.4. The gradient itself is simply copied from the interior cell I to the ghost cell H so that the gradient is only indirectly imposed to be unitary.



(a) Upwind location (wall)

(b) Downwind location (farfield)

Figure 3.4 Ghost cell filling through extrapolation method for two scenarios

Note that the solid surfaces where the icing velocity is null, meaning that there is no ice

accretion, can be conserved by using the extrapolation method at the boundaries. In fact, it is not possible to directly impose a solid boundary in the level-set temporal evolution. However it can be done indirectly through the velocity field with a null velocity. This is covered in §3.3.2.

3.2.4 Eikonal equation

To initialize the variable ϕ before any unsteady simulation is performed, the Eikonal equation (Eq. (3.6)) is solved. It is a variant of the level-set equation featuring a unit source term which is used to compute the wall distance with a PDE formulation [111]. In its minimal form, it forces the norm of the gradient of ϕ to be unitary, which is a property of a signed distance function. By squaring the equation and adding a pseudo-time derivative which vanishes at steady state, the convective form of the Eikonal equation is obtained and can be solved with the same spatial schemes used for the level-set equation. Only the implicit temporal integration methodology is reused, as the dual time-stepping technique is not necessary for a steady equation. A main difference is that a Dirichlet boundary condition is applied at the wall to enforce a null distance on these boundaries.

$$\begin{aligned} |\nabla\phi| &= 1 \\ \frac{\partial\phi}{\partial\tau} + \mathbf{n}_\phi \cdot \nabla\phi &= 1 \end{aligned} \tag{3.6}$$

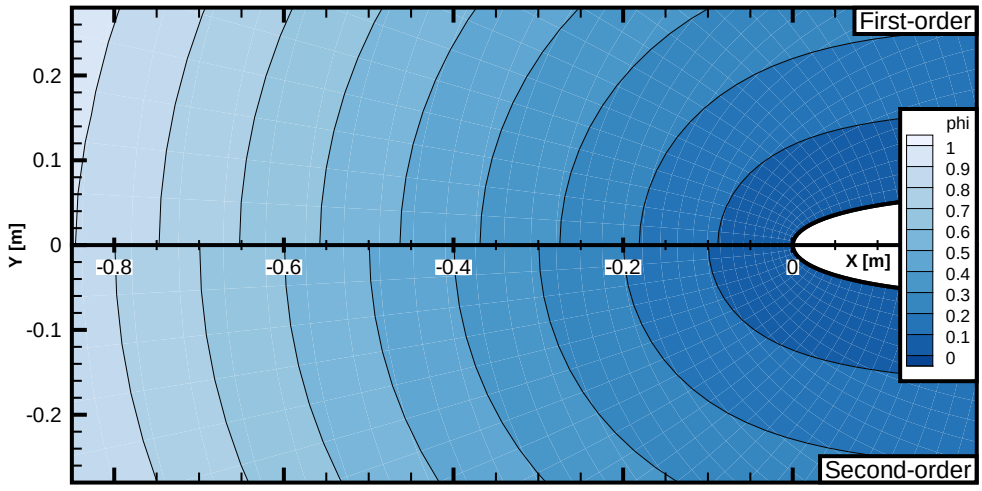


Figure 3.5 Impact of the spatial accuracy on the Eikonal solution

Figure 3.5 show two simulations performed on a NACA0012 symmetric airfoil. Each solution

is symmetrical along the X-axis and only one half is shown : i) the first-order accurate result above the axis and the second-order accurate result below the axis. The contour colours and lines represent ϕ which is the wall distance computed by the Eikonal equation (Eq. (3.6)). In the case of the second-order accurate solution, the intersection of the contour lines with the X-axis shows clearly that the wall distance is accurately computed, whereas for the first-order case the solution is offset. In fact, the gradient norm average in all the field is around 1.05 for the first-order accurate solution compared to 1.001 for the second-order solution. Recall that the gradient norm should converge to 1. Furthermore, the grid used is of high quality and the case itself is simple with a symmetric airfoil. With lower grid quality and more complex geometries, the first-order scheme yields solutions with a higher degradation.

3.2.5 Re-initialization equation

According to Osher and Fedkiw [68], the temporal evolution of the signed distance function ϕ is accurate near the interface, but is subject to degradation away from it. Although only the interface solution is important, this degradation can compromise the resolution of the global system by deteriorating the matrix conditioning. To solve this issue, re-initialization of the signed distance solution is required. However, such a method must be considered with care, as the interface should not be affected by the re-initialization [4].

Therefore, after several physical time cycles of the level-set unsteady resolution, Eq. (3.7) is solved to perform this re-initialization step. The same numerical methodology presented earlier is used to solve Eq. (3.7) in a steady manner. The boundary conditions are identical to those used for the level-set equation : an extrapolation with a unitary gradient to the ghost cells.

$$\frac{\partial \phi}{\partial \tau} + H\left(\phi\right) \mathbf{n}_{\phi} \cdot \nabla \phi = H\left(\phi\right) \tag{3.7}$$

In Eq. (3.7), $H\left(\phi\right)$ is a smeared Heaviside function which is continuous in the range -1 to 1 depending on the sign of ϕ. This function is evaluated as :

$$H\left(\phi\right) = \frac{\phi}{\sqrt{\phi^2 + \varepsilon^2}} \tag{3.8}$$

where ε is a local cell characteristic length. This smeared Heaviside function is near zero in a band of width ε around the interface and goes rapidly to 1 or -1 to apply the re-initialization only in the regions away from the interface. Furthermore, it reverses the sense of the gradient

so that the negative region is downwind to the interface. The interface band is acting as ghost cells for the re-initialization of the positive and negative regions.

3.3 Icing velocity definition

The previous sections presented diverse manners to solve the signed distance function ϕ either in an initial state, a moving state or a re-initialized state. This section focuses on another variable: the external velocity field $\mathbf{V}$ convecting the interface. In this context, it is called icing velocity and is noted $\mathbf{V_i}$. Given that ice grows in the normal direction from a surface, this velocity is proportional to the interface normal ($\mathbf{n}_\phi$) and its norm is driven by the wall ice mass accretion rate computed in the thermodynamic module. Thus, its exact initial value at the wall is defined in Eq. (3.9) :

$$\mathbf{V_i}_{wall} = \frac{\dot{m}_{ice,wall}}{\rho_{ice}\Delta S_{wall}}\mathbf{n}_{\phi_{wall}} \tag{3.9}$$

where ρ_{ice} is the ice density and ΔS_{wall} is the area of a surface element on the wall.

Pena *et al.* [70] suggested that the wall icing velocity should be directly propagated in the field from the initial geometry. However, as $\mathbf{V_i}$ contains a local area component ΔS, its value should also evolve due to the local stretching or shrinking of the interface during the simulation and this is related to the local volume cell dimensions. Hence, another velocity profile is proposed in Eq. (3.10) which allows to consider the local cell dimensions in relation to the interface orientation to recover a local icing velocity from the ice mass accretion rate $\dot{m}_{ice}$ which is propagated from the wall in the domain.

$$\mathbf{V_i}_{local} = \frac{\dot{m}_{ice,local}}{\rho_{ice}\Delta S_{local}}\mathbf{n}_{\phi_{local}} \tag{3.10}$$

In Eq. (3.10), ΔS_{local} is a cell characteristic dimension projected on the interface computed with Eq. (3.11) in which $\mathbf{n_f}\Delta S$ is an element's facet normal scaled with the facet area. The projection of the scaled facet normals on the interface is summed over all facets of a cell and then divided by 2 so to average the contribution of each topological direction. The use of a local velocity allows the level-set process to conserve the swept volume as opposed to the approach of Pena *et al.* [70] where the wall velocity allows to conserve the swept distance. Note that both would yield the same results if the interface is perfectly linear, such as on a

flat plat with constant uniform ice accretion rate over it.

$$\Delta S_{local} = \frac{1}{2} \sum_{facets} |\mathbf{n_f} \Delta S \cdot \mathbf{n}_\phi| \tag{3.11}$$

3.3.1 (Re)-Propagation equation

The icing velocity needs to be defined in the field from the solid surfaces. It is initialized at zero and it depends on the variables located at the wall, either $\mathbf{V_i}_{wall}$ or $\dot{m}_{ice,wall}$. Different methods exist to perform this extrapolation operation, generally categorized as: i) a search algorithm, ii) a convection equation or iii) a fast marching method [68]. The convection equation is particularly interesting, as mentioned by Frolkovič *et al.* [37], as it can be formulated in the same way as the other equations previously presented: the level-set, the Eikonal and the re-initialization. Furthermore, by reusing the Heaviside function (Eq. (3.8)) introduced for the last, a two-way extrapolation is possible from the interface region to perform re-propagation of the icing velocity depending on its value at the current interface location. This allows to adapt the velocity field to the change in the interface normal.

The variable propagated from the wall into the domain (φ) is either associated to $\mathbf{V_i}_{wall}$ or $\dot{m}_{ice,wall}$ depending on the chosen icing velocity formulation between Eq. (3.9) or Eq. (3.10). Eq. (3.12) is discretized in a similar manner as the level-set equation, but the time-accurate part is removed as the propagation equation is solved in a steady state.

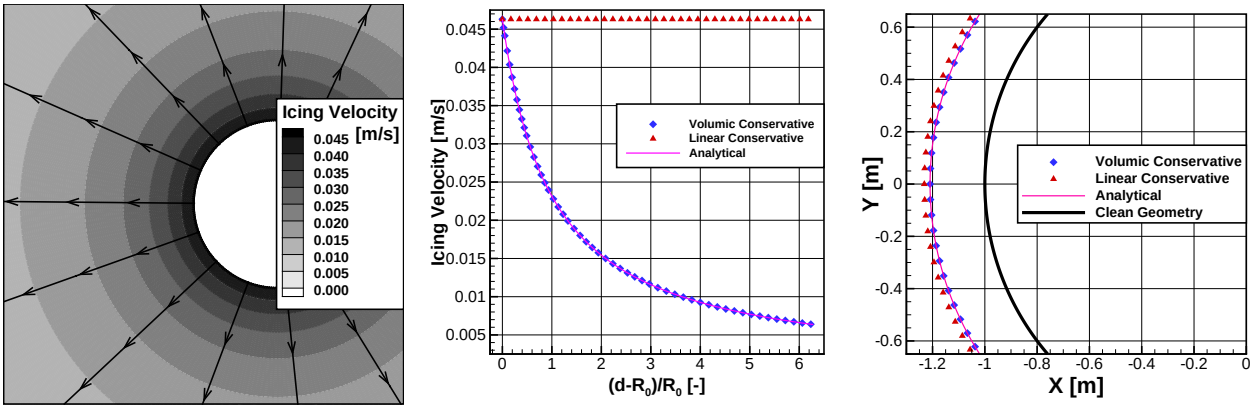
$$\frac{\partial \varphi}{\partial \tau} + \nabla \cdot (\varphi \mathbf{n}_\phi) - \varphi \nabla \cdot \mathbf{n}_\phi = 0 \tag{3.12}$$

In Eq. (3.12), τ is the pseudo-time. This equation should also contain the Heaviside function in the convective fluxes part as seen in Eq. (3.7), but it was found that only the icing velocity in the positive ϕ section needs to be updated, since the negative section corresponds to the new ice and is kept constant throughout the rest of the simulation. Furthermore, the treatment of the cells around the area where the moved interface joins the geometry is troublesome and might lead to failures in the re-propagation of the icing velocity. Therefore, the deactivation of its computation in the negative region avoids potential issues and also preserve the already frozen region. Furthermore, the deactivation is extended in the positive narrow band around the interface defined by the Heaviside function.

3.3.2 Boundary conditions

The physical boundary conditions applied to the propagation equation (Eq. (3.12)) are slightly different than those applied to the level-set equation. The farfield is treated with a Neumann boundary condition with zero variation along the gradient direction and the wall is treated with a Dirichlet condition with the known ice mass accretion rate. The latter is crucial in the conservation of the solid geometry shape where there is no ice accretion. When $\dot{m}_{ice,wall}$ goes below a low number ε, it is replaced by 0 to avoid numerical errors. In this work, ε is set to 1 µg of ice. It is then extrapolated in the field normal to the initial geometry and a portion of the zero velocity initial condition is conserved.

3.3.3 Impact of velocity type



(a) Cell dependent velocity in the field

(b) Comparison of the velocity on a radial line

(c) Comparison of the iced solution

Figure 3.6 Impact of the icing velocity formulation on the velocity and icing solutions

In order to highlight the impact of the icing velocity choice, a physical phenomenon is presented where a cold pipe with a circular section is immersed in unfrozen water, thus initiating an ice accretion phenomenon. The arbitrary user-defined parameters make this test suitable for verification as a canonical case. A cylinder with a unit radius is meshed uniformly in cylindrical coordinates with 256 cells in each grid direction and a constant ice accretion rate of 1kg/s is applied uniformly on each facet of the surface mesh. A constant time step of 0.5 seconds is chosen for an overall simulation time of 5 seconds.

The local icing velocity field is presented on Figure 3.6(a), where the reduction of velocity due to an increase in the field cell dimensions is apparent. The trivial icing velocity field solution obtained for the wall velocity approach is uniform and is not shown. Since the analytical solution depends on the exact circumference, a radial line is extracted for each

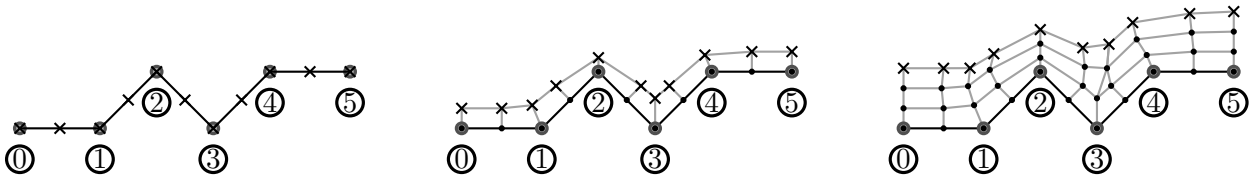
solution and presented on Figure 3.6(b) where it can be seen that the local velocity approach yields a result that is close to the analytical solution. The horizontal axis represents the normalized distance d from the edge of the cylinder of radius R_0 with the radius acting as a reference value for the normalization. The maximum error in this case is 0.01%. In Figure 3.6(c), the analytical solution of the ice accretion is compared to the results obtained with both velocities and it can be seen that the volume accumulated using the extrapolated wall velocity is higher compared to the solution obtained with the local velocity, as expected. The relative error with respect to the analytical ice mass is 11.6% and -0.03% respectively for the wall velocity and the local velocity formulations. From this point onward, the level-set simulations use the local icing velocity computed as described in Eq. (3.10).

3.4 Explicit tracking

In order to achieve a global ice mass conservation, an explicit Lagrangian algorithm is added to track the zero contour level in the domain. Although it might seem contradictory to combine an explicit method with an implicit surface definition, this Lagrangian algorithm is used only to follow the implicit interface, not to define its movement. The issues mentioned in §3.1 caused by explicit methods on the final geometry definition are thus not as important.

A number of seed points are created on each facet of the geometry after the solution is initialized with the Eikonal equation, as seen on Figure 3.7(a). By making use of the already computed level-set gradient which describes the direction toward where the interface is moving, the path of the tracking points are integrated iteratively, as seen in Figure 3.7(b), until they reach the interface. At each movement of the points, the variables required for the path integration are interpolated from the field solution. In the case where the point goes past the interface, the final position of the tracking point is interpolated based on the values of the ϕ variable at the previous and actual locations in order to be exactly on the zero contour level. The volume between the previous state of the tracking points, marked with small black dots in Figure 3.7, and the actual state marked with crosses is computed by considering the points to be connected together in a structured way which generates quadrilateral cells. The virtual connection is shown with light gray lines in Figure 3.7.

The points are moved using small and adaptive steps that prevent overlaps and negative volumes. To further add robustness, if the distance between two points becomes small, the edge between the points is collapsed, as seen on Figure 3.7(c) around geometry vertex 3, and one of the nodes is forced to follow the other. That node is named a slave point and it might be freed again later in the process if the distance to the next adjacent point is increasing beyond a certain limit which might happen if an initial concave region is filled and then



(a) Initialization (b) Normal growth (c) Nodes merged at 2^{nd} iteration

Figure 3.7 Main steps of the explicit tracking process

becomes convex. As seen on Figure 3.7(c), a collapse of a virtual edge generates a triangle instead of a quadrilateral cell, but it avoids the creation of negative volumes.

A major advantage of this process is that the global volume swept by the interface becomes known during the level-set evolution process and can therefore be used as a stopping criterion for the simulation when the predicted ice volume is reached up to a tolerance, allowing to globally conserve the mass. As the level-set equation models the ice accretion in a time accurate manner, the full physical icing time should be reached during the simulation. However, it has been found that the global mass conservation stops most of the simulations before the full time is reached, which hints that a local conservation correction might be necessary to better adapt to the interface changes in length and in shape. Another advantage is that the tracking points can be directly used to extract the interface coordinates at the end of a layer simulation to perform a B-spline approximation, which saves the need to extract the interface from the field solution with another algorithm.

3.5 Global conservative level-set process

As seen in the previous sections, the level-set computation process includes four different equations which are solved alternately: the Eikonal (Eq. (3.6)), icing velocity (re-)propagation (Eq. (3.12)), level-set (Eq. (3.1)) and re-initialization (Eq. (3.7)) equations. Figure 3.8 describes the order in which the four equations are solved. The goal of the re-propagation and re-initialization equations is to improve the ice accretion behaviour of the level-set simulations and its ice mass conservation. As such, the complete process can be called a conservative level-set approach. However, these two equations do not need to be called at each level-set cycle, which explains the multiple loops in the process.

In the flowchart of Figure 3.8, iLS represents the number of level-set cycles, iLS_{max} represents the number of level-set cycles between each call to the re-initialization and propagation equations, t is the physical time and Δt_{ice} is the icing time for the layer. The volume swept by the interface is noted Ω and the ice volume to obtain is noted Ω_{ice}.

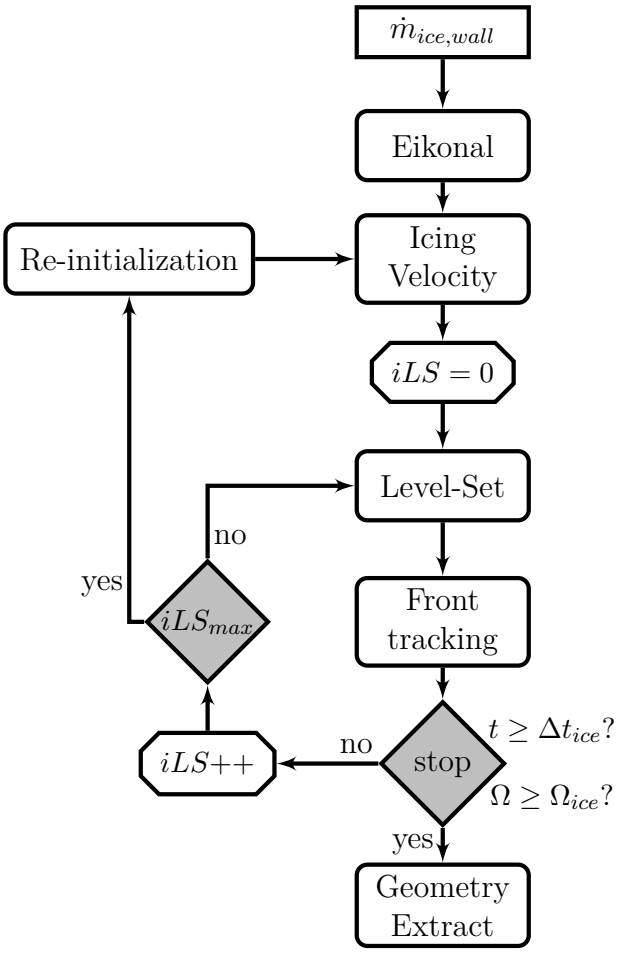


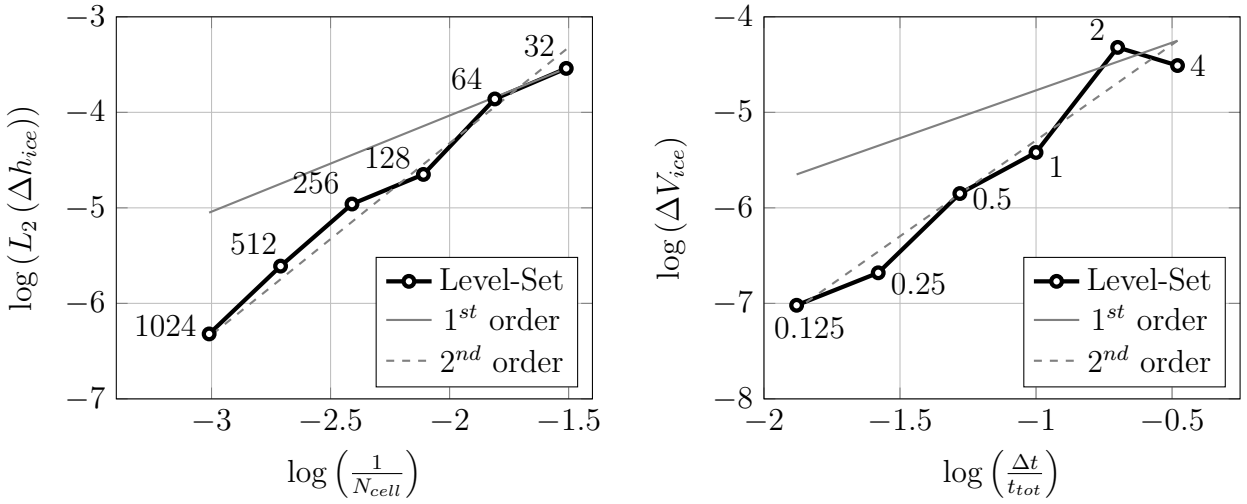
Figure 3.8 Conservative level-set process

3.6 Space-time accuracy study

In this section, a grid convergence study and a time convergence study are performed to verify the accuracy order reached with the spatial and temporal discretization schemes used to solve the different equations of the level-set process. In both cases, the simulations are run without re-propagation of the icing velocity and re-initialization of the solution for predicting an analytical interface solution as the icing velocity in the domain then depends only on the initial geometry which is defined analytically. The accuracy of the simulation is assessed with a L_2 norm of the difference between the 0-level contour and this analytical solution.

The second-order spatial accuracy is verified on a unit square box with 2 sides being defined as solids with a total ice mass accretion rate of 64 $\mathrm{kg\,s^{-1}}$ by side, which is split uniformly on each solid facet and the physical simulation time corresponds to 5 seconds. A family of Cartesian meshes is produced from a fine grid made of 1024 by 1024 cells by reducing by half the number of cells down to 32 cells in each grid direction for each coarser grid. The

resulting ice front is an L shape which is symmetric with respect to the $y = x$ line. Its analytical position after 5 seconds is on the lines $y = (4/11)\,\mathrm{m}$ and $x = (4/11)\,\mathrm{m}$.



(a) Grid convergence study

(b) Time convergence study

Figure 3.9 Results of the (a) spatial and (b) temporal accuracy order verification

Figure 3.9(a) shows the evolution of the error with respect to the mesh dimension, with the axis in logarithmic form. The results are compared to the trends of first- and second-order accuracy. The final order of accuracy using the three finest meshes is computed [102], yielding an order of accuracy around 2.1.

In order to verify the second-order temporal accuracy of the level-set solver, a cylinder case with uniform ice accretion is chosen. The cylinder has a unit radius and an ice mass accretion rate of 1kg/s is defined on each facet of the surface mesh, as described previously in §3.3.3. The grid is made of 256 cells in each I- and J-directions and the spacing between each J-layer is constant, with the first cell height being equal to the size of the surface facet. The physical simulation time is 5 seconds and the physical time step is varied through a physical CFL number going from 4 to 0.125 acting as a scaling factor. In this case, the analytical solution is found by computing the ice volume which should be accumulated and adding this volume to the initial cylinder to obtain a new radius of 1.2095m.

Figure 3.9(b) shows the evolution of the error with respect to the time step and the comparison with the trends of first- and a second-order accuracy. The results are non-monotonic due to a probable influence of the mesh and the interpolation method. Therefore the exact order of accuracy cannot be computed as it was done for the spatial accuracy, but a qualitative comparison with the second-order trend line allows to conclude that the second-order accuracy in time is reached. This non-monotonicity might be caused by numerical errors as

the absolute value of the error is very low and involves the computation of squared values which become close to the machine accuracy.

3.7 Results

In this section, a comparison between a manufactured geometry is first shown to highlight how the level-set performs in concave regions compared to an algebraic method. Two cases on a NACA0012 airfoil are then presented and compared to experimental data [100]. These cases are tested with different number of ice layers to show ice shape convergence in the multi-layer quasi-steady approach. All level-set simulations use a case dependent physical time step which leads to a different number of physical dual time stepping cycles, each cycle made of 50 pseudo-time iterations to converge the temporal evolution to a residual error decreased by 3 orders of magnitude. For the NACA0012 cases, the initial meshes have 385 by 257 vertices in the grid directions, with an increase in number of I-vertices at each layer depending on the length of the ice shape and its curvature. Table 3.1 presents the parameters of the different icing cases. Note that the wall roughness used in the simulations are computed through a semi-empirical model from Shin *et al.* [88]. The chord of all geometries is $0.5334m$.

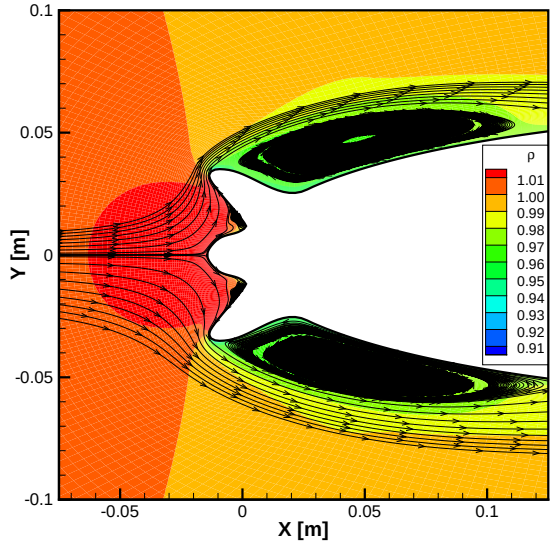
Table 3.1 Level-set cases parameters

Case		**Manufactured**	**Rime Case 1**	**Glaze Case 3**
AoA	[°]	0.0	4.0	4.0
Mach	[−]	0.177	0.325	0.317
Temperature	[K]	269.10	250.70	262.25
Pressure	[kPa]	95.610	101.325	101.325
LWC	[g/m^3]	1.30	0.55	1.80
MVD	[µm]	20	20	30
Icing Time	[s]	480	420	360
Roughness	[µm]	3550.6	618.7	3976.5

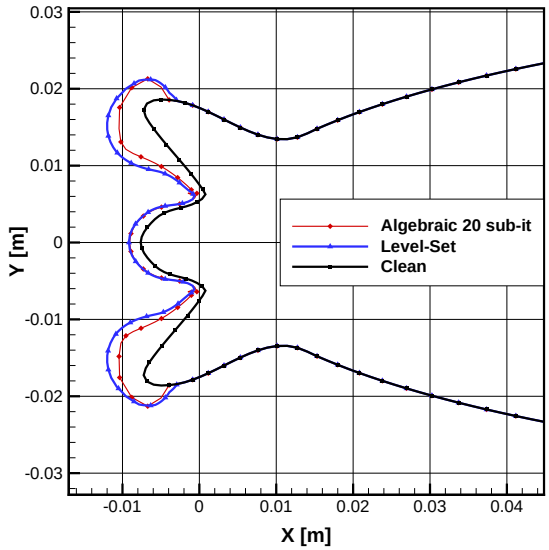
3.7.1 Manufactured Case

This first case is simulated on a manufactured airfoil. The purpose of this test is to compare the level-set solver to the algebraic iterative approach presented in §2.5.1. The geometry, first presented by Lavoie *et al.* [52], is built by adding three ice-like horns on a symmetric NACA0012 airfoil, thus generating four recirculation zones and multiple stagnation points in the flow solution. Four water accumulation points are generated as the runback water modelling depends only on the air shear stress in NSCODE-ICE. This makes it possible to study the impact of concave regions on the geometry evolution. The recirculation zones can

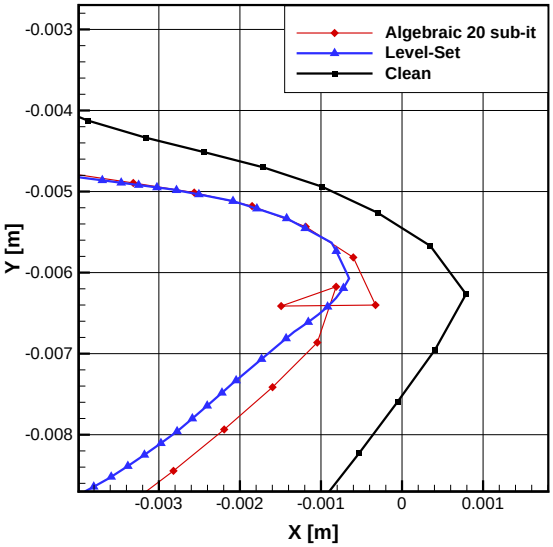
be seen on Figure 3.10(a) where the contour of the air density is presented along with airflow streamlines.



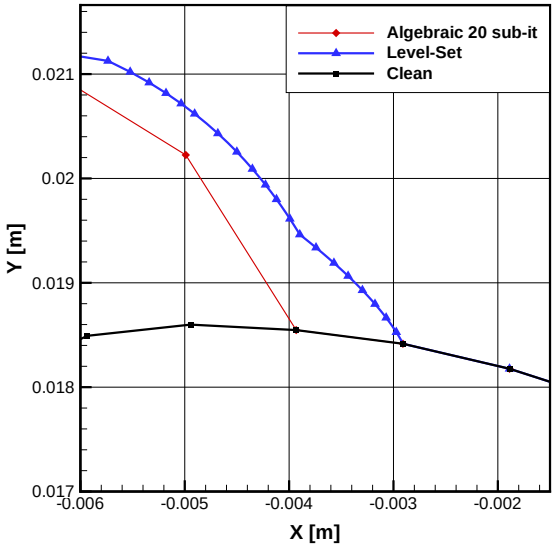
(a) Recirculation zones



(b) Ice accretion for manufactured case



(c) Close view on lower concave region



(d) Close view on ice region limit

Figure 3.10 Manufactured results for level-set and algebraic methods. Figures adapted from Bourgault-Côté *et al.* [16]

The ice accretion results comparing the level-set to the algebraic method are presented in Figures 3.10(b) to 3.10(d). Note that re-propagation and re-initialization are not performed in order to best compare with the evolution behaviour of the algebraic method which depends only on the initial surface mesh configuration. On a global scale, both methods lead to similar results. However, as shown in the close view of Figure 3.10(c), the algebraic method contains

an overlapping geometry in the concave region. This is consistent with the findings of Pena *et al.* [70]. In this case, a multi-layer icing simulation would be impossible to perform when using the algebraic models without a method to repair the solution [84] as the new geometry could not be used to obtain a new mesh or to move the existing mesh. On the other hand, the level-set method leads to a valid geometry and a notable feature of its solution is the smoother limit definition at the end of the iced region (see Figure 3.10(d)), due to its high density of points defining the front solution. This is due to the fact that the level-set method is solved in the volume field, which is refined in the normal direction from the wall for the flow simulation, whereas the algebraic method is limited in spatial discretizations by the number of elements on the initial surface mesh.

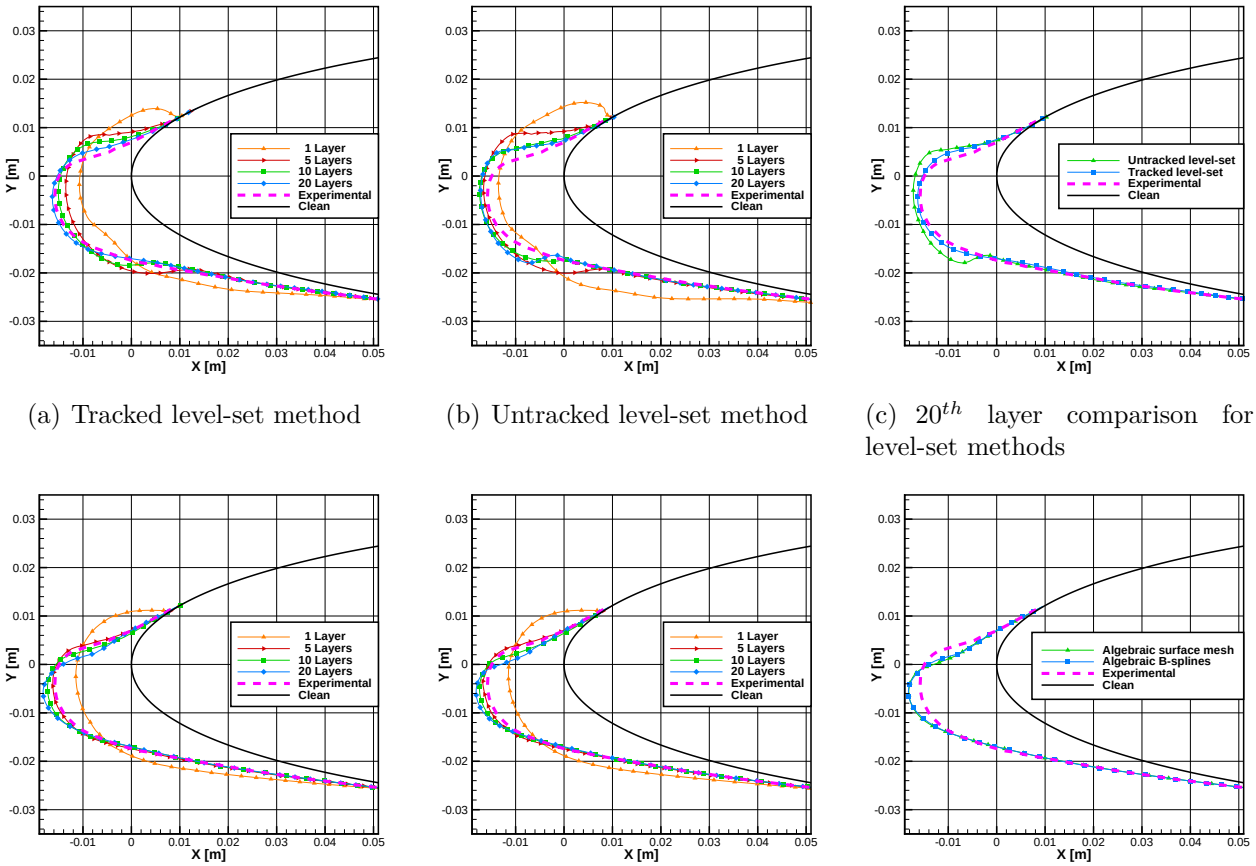
3.7.2 Rime Ice

The first experimental case is a rime ice case which corresponds to "CASE_01" in Trontin *et al.* [100]. It was run in 1, 5, 10 and 20 layers with four different methods : i) the tracked level-set method described in §3.5, ii) the untracked level-set method, iii) the algebraic method described in §2.5.1 using B-splines to regenerate its geometry between layers and iv) the algebraic method using the displaced surface mesh directly to regenerate a new mesh. The second method is named untracked level-set because the explicit tracking is deactivated, which does not allow to compute the actual swept volume and to control the mass conservation globally. Table 3.2 summarizes the different components of the previously listed methods.

Table 3.2 Rime case methods combination

Method	**Ice accretion**	**Geometry**	**Mass control**
i	Level-set	B-splines	Explicit tracking
ii	Level-set	B-splines	No
iii	Algebraic	B-splines	No
iv	Algebraic	Surface mesh	No

Figure 3.11(a) shows a smaller ice volume for all multi-layer tracked level-set simulations when compared to Figure 3.11(b). In this case, the tracked level-set simulations converge towards the experimental shape better than the untracked level-set simulations and Figure 3.11(c) better highlights the difference by showing only the 20 layers results. This is caused by the use of the volume global conservation stopping criterion from the explicit interface tracking method which stops the simulation when the desired ice volume is reached. Moreover, Table 3.3 presents the total error for all layers between the desired volume and simulated



(a) Tracked level-set method

(b) Untracked level-set method

(c) 20^{th} layer comparison for level-set methods

(d) Algebraic method with B-splines

(e) Algebraic method with surface mesh

(f) 20^{th} layer comparison for algebraic methods

Figure 3.11 Results for the rime ice case "CASE_01" from Trontin *et al.* [100]. Figures adapted from Bourgault-Côté *et al.* [16]

volume of the tracked level-set simulations, which is almost negligible compared to the overall error of the multi-layer simulations introduced by the geometry approximation and mesh regeneration.

By comparing Figure 3.11(a) to Figure 3.11(d), it can be seen that the level-set approach is in better agreement with the experimental shape than the algebraic approach. Although both methods are converging in layers, two different solutions are obtained. The discrepancy in the algebraic method result is due to : i) the use of the bisectors between two neighbour facets from the surface grid to displace the vertices and ii) the limitation in the range of step size where too high a value results in wrong shapes and too low a step size results in self-induced oscillations of the surface mesh. On the contrary, the level-set method directly uses the interface normal ($\mathbf{n}_\phi$) to move the level-set front at each step of the computation, which better represents the ice accretion direction as it is computed from a gradient method

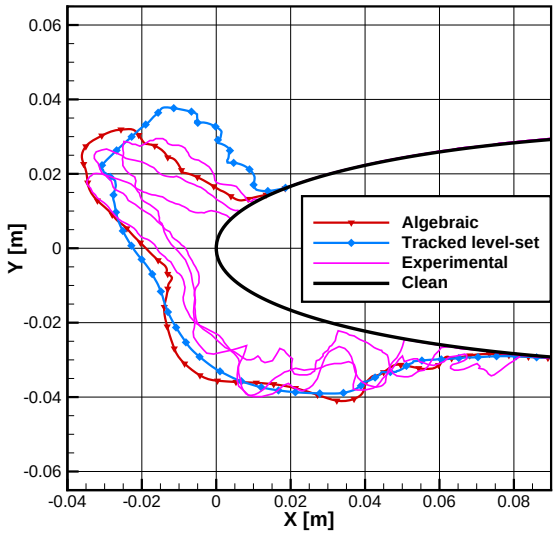
using an extended stencil including all facet neighbours and vertex neighbours. Furthermore, the unsteady nature of the level-set method requires small physical time steps which allow for considering a smoother evolution of the interface normal. Another observation highlighting a weakness in the physical modelling of the algebraic method is that it yields a better agreement with the experimental data when using 5 layers instead than 20 layers.

To study the effect of using the B-splines instead of the displaced surface mesh in the algebraic method, Figure 3.11(d) is compared to Figure 3.11(e), with the 20 layers result shown in Figure 3.11(f). The solutions produced with B-splines are smoother, as expected, but the main difference lies in the degradation of the process automation when using the surface mesh. Depending on the user definition of the smoothing coefficients used by the algebraic method, invalid geometries can be obtained at any layer in the simulations. Compared to this issue, the B-splines are smoothing appropriately all the unphysical self-enhanced oscillations in the solutions of the intermediate layers without user intervention. Overlapping might still occur, as it is an explicit algebraic method, but it is less likely in a rime case where highly concave regions are uncommon.

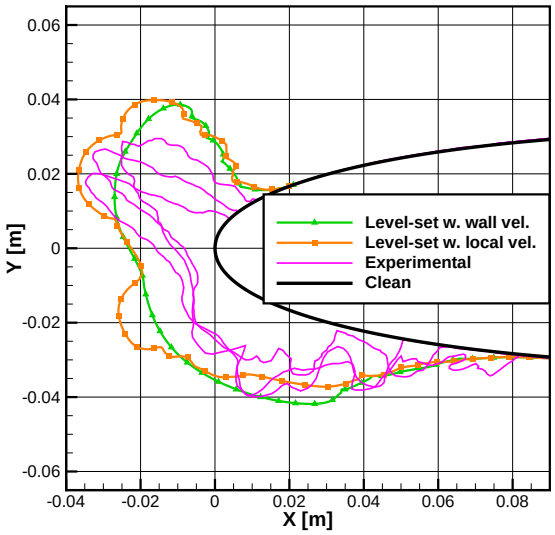
3.7.3 Glaze Ice

The second experimental case is a glaze ice case which corresponds to "Condition No. 3" in Trontin *et al.* [100]. The purpose of this study is to show how the level-set algorithm performs on a more complex case and also to show the effects of the global conservative limiter in the explicit swept volume tracking method. Furthermore, as a glaze ice shape features concave regions, it is suitable to study the effect of using a local icing velocity instead of the wall velocity. It was run in 1 and 5 layers only as the ice horn is more difficult to mesh using an elliptic structured grid generation algorithm, leading to a higher probability of mesh failure when using more layers. Four methods are compared in this case : i) the tracked level-set approach, ii) the untracked level-set method, iii) the untracked level-set method using the wall icing velocity described in §3.3 and iv) the algebraic method using B-splines. The results are compared to three different experimental shapes which represent the data envelope, highlighting the large experimental uncertainty.

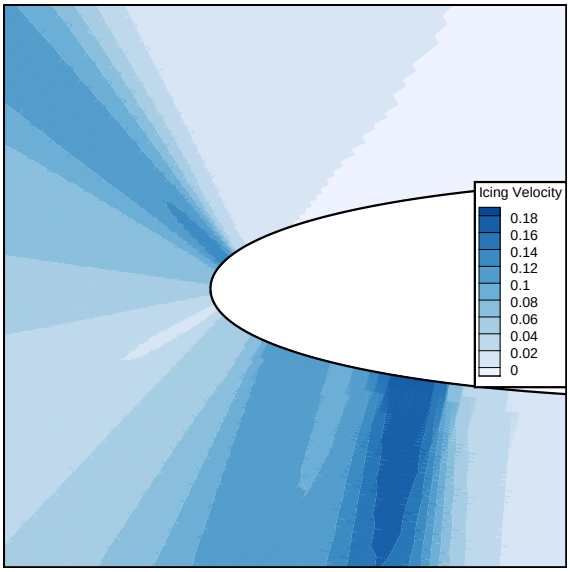
As it can be seen in Figure 3.12(a), the tracked level-set and algebraic methods are both able to capture the lower part of the ice shape. Looking at the ice horn, we see that the algebraic method is in better agreement with the experimental data in terms of the width. Actually, the higher width of the level-set solution indicates a deviation from the physical behaviour which probably occurred at the first layer as it is the most critical moments of the simulation where a small difference in ice shapes leads to a different position of the main ice



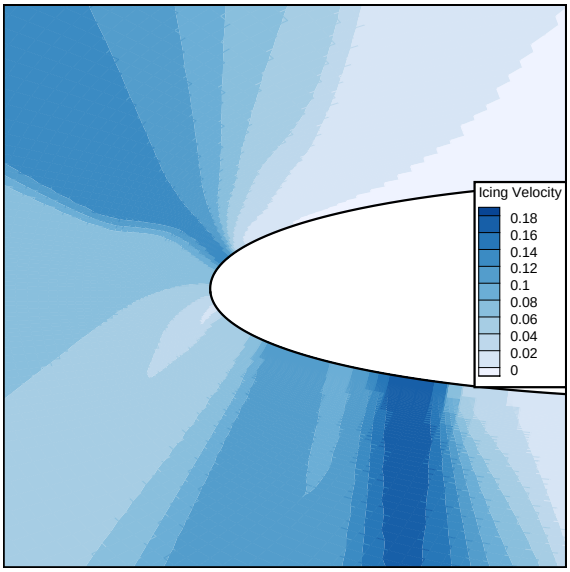
(a) Level-set compared to algebraic for 5 layers



(b) Icing velocity effect for 5 layers



(c) Wall icing velocity at 1^{st} layer



(d) Local icing velocity at 1^{st} layer

Figure 3.12 Results for the glaze ice case "Condition No. 3" from Trontin *et al.* [100]. Figures adapted from Bourgault-Côté *et al.* [16]

horn. The horn is also larger than the experimental shapes, but adding layers would reduce the width in the process of converging toward a solution, as seen in Chapter 6. Nevertheless, the tracked level-set result is in better agreement than the results obtained with the two untracked level-set methods presented in Figure 3.12(b), which indicates that the explicit interface tracking allows to conserve the global ice mass.

The effect of the icing velocity can be seen on the final ice results in Figure 3.12(b) and in the

field as computed to obtain the first layer in Figures 3.12(c) and 3.12(d) for the untracked level-set method using respectively the wall icing velocity and the local icing velocity. The converged ice results obtained with the local icing velocity show a better qualitative agreement with the experimental shapes.

3.7.4 Error Analysis

By having the geometry kernel EGADS integrated in the software, it is straightforward to compute the exact internal volume of the geometry after each layer of ice as a continuous geometry is defined through B-splines. Also, the exact volume swept by the level-set interface is known when using the explicit level-set tracking method. Thus, knowing how much ice mass should have been accreted at each layer by extrapolating the surface ice accretion rate, $\dot{m}_{ice}$, with the icing time, two error values can be computed. First, the ice mass error of the level-set simulation, named tracking error, can be computed from the tracked swept volume and the extrapolated ice volume, thus characterizing the conservation error of the solver. Second, the global volume error induced by the B-spline reconstruction can be computed by comparing the final geometry internal volume to the extrapolated ice volume. The method using EGADS can also be used with the results computed with the algebraic method using B-splines to provide a global volume error, but not for the method using the surface mesh which is therefore not included in the table. Note that for multi-layer simulations, the real and predicted ice mass are summed separately at each layer before computing the errors in order to remove the error caused by the re-discretization and re-meshing steps. Table 3.3 presents the percentage of gain or loss in ice mass for each case.

Table 3.3 Ice mass error

Case	Method	1 Layer	5 Layers	10 Layers	20 layers
Rime	Tracked level-set : Tracking	0.03%	0.01%	0.02%	0.04%
	Tracked level-set : Global	0.01%	−0.27%	−0.81%	−1.30%
	Untracked level-set	25.80%	14.90%	8.67%	12.00%
	Algebraic with B-splines	0.98%	0.32%	0.14%	−0.16%
Glaze	Tracked level-set : Tracking	0.01%	0.02%	-	-
	Tracked level-set : Global	0.58%	1.35%	-	-
	Untracked level-set w. wall vel.	24.40%	8.34%	-	-
	Untracked level-set w. local vel.	34.60%	18.30%	-	-
	Algebraic with B-splines	1.19%	0.28%	-	-

It can be seen that all the cases solved with the tracked level-set have error values below 0.05 % for the volume swept by the level-set during its computation. Furthermore, this low

error is independent from the number of layers used in the simulation. However, as the final level-set solution is converted to B-splines, an error on the ice volume is introduced at each layer, thus increasing the global error when more layers are used. The sign of the error is also case dependent, as there is a loss of ice mass in the rime case, but a gain in the glaze case. In all cases other than untracked level-set, the absolute global error value is below 1.5 %. This is acceptable since experimental data variability is higher, as shown in a recent benchmark study of ONERA's icing software IGLOO2D [100]. For the untracked level-set methods, the high errors stem from the absence of any conservation criterion and these errors are the reason behind the addition of the explicit tracking method allowing to control the conservation of the ice volume.

3.8 Conclusion

In this chapter, a mass conservative level-set algorithm that models the geometry evolution for icing is presented. The complete process includes the sequential resolution of four equations that play different roles: i) the Eikonal to initialize ϕ, ii) the (re)-propagation to extrapolate $\mathbf{V_i}$, iii) the level-set to evolve the ice front and iv) the re-initialization to repair the degradation caused by the front evolution in ϕ. The numerical developments which are used by the four equations are detailed for the level-set with an emphasis on the spatial and temporal discretization schemes which are then verified on grid and time convergence studies. Furthermore, two formulations for the icing velocity ($\mathbf{V_i}$) are compared and the velocity based on local cell dimensions [16] shows a better agreement with the ice accretion physical behaviour than the one used by Pena [72].

An explicit tracking method is also added to better control the ice mass conservation and the effects are shown in the icing results where the global ice mass error is reduced by one order of magnitude when the tracking is active. The icing cases simulated include a manufactured case, a rime ice case and a glaze ice case. The first acts as a comparison case on how the level-set performs in concave regions where an algebraic method fails to deliver a valid geometry. The two other cases are compared to experimental data and show that the level-set results converge better toward the experimental data than the algebraic results. It benefits from using the RANS mesh for more accurate near-wall discretizations, as compared to the surface mesh. Furthermore, the volume error caused by the tracked level-set is negligible (below 0.05%) and the global process error is below 1.5% in all cases, which is acceptable compared to experimental variations as seen on Figure 3.12(a). Finally, the results and comparisons presented in this chapter show that the level-set method is an alternative to algebraic methods by providing a high-fidelity PDE-based solution to the ice front evolution.

CHAPTER 4 HYPERBOLIC ICE ACCRETION

In Chapter 3, the level-set equation was used to capture implicitly the air-ice interface. Although this approach has advantages, its dependency on the volume field and its high computational cost reduce its application potential for 3D simulations. An original method, based on the hyperbolic mesh generation technique discussed in §2.7.4, is developed to solve these issues. This chapter presents the adaptation of the hyperbolic mesh equations for ice growth and various canonical cases to highlight the capacity of the hyperbolic ice accretion method. Results on experimental cases with mass conservation are then shown.

4.1 Rationale

The geometry evolution is a process that depends on the upwind direction aimed outward from the geometry and in fact orthogonal to it by hypothesis in this work. A suitable formulation for this process thus requires either parabolic or hyperbolic methods, such as the algebraic method or the level-set method. The upwind direction can be seen on Figure 3.6(a) where the direction of the icing velocity ($\mathbf{V_i}$) is illustrated with stream traces starting at the cylinder surface and pointing outward. These stream traces are obtained with the level-set method and they illustrate that an implicit method with a local formulation can still depend on the global interface to define the upwind direction by being defined in a higher topological dimension.

This continuous definition of the interface displacement direction explains in part the quality of the ice shapes definition with the level-set method. However, the level-set has a high computational cost as a drawback, due in part to its definition in the complete volume field rather than only on the surface. Furthermore, the use of grids generated for a RANS solver, which are fine in the boundary layer region near the surface, forces the use of a very low physical time step for the unsteady evolution of the level-set equation resulting in a high number of iterations to obtain the final ice shape. Overall, level-set computations can last from 4 to 20 times the flow computations in a multi-layer icing simulation depending on the case. This drawback is not studied in Chapter 3 since efficiency is not the focus of this study and no effort is invested to accelerate the level-set resolution. Nevertheless, an efficient method is a requirement for a realistic 3D extension and such a high computational cost is not satisfactory.

In contrast, the algebraic (Lagrangian) methods present a high efficiency with negligible

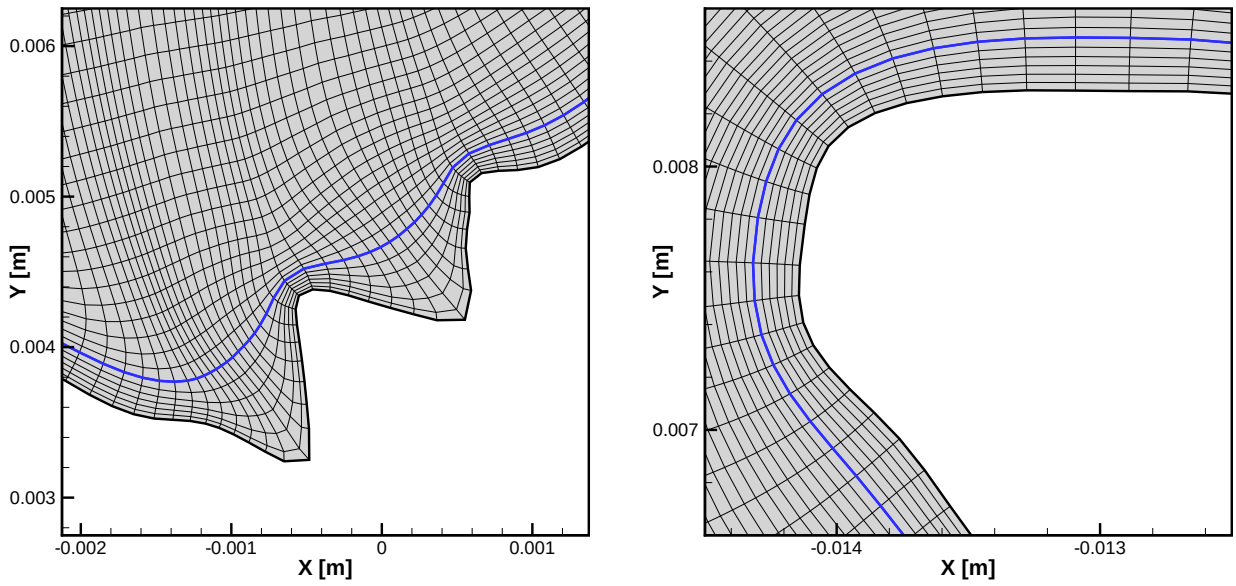
computational cost, but it was already stated that their robustness is low. This kind of explicit method being limited to a local surface definition, the global knowledge of the interface is not available and surface crossing can happen with non-neighbouring portions of the surface. It thus requires various heuristic correction methods to limit the risk of obtaining invalid geometries, such as grid folding detection, sharp angle removal, thin concave region filling. Furthermore, their definition is quite far from the desired PDE-based method and the extension to 3D is not straightforward.

From these observations, a new method combining robustness and efficiency is sought for a viable 3D extension to model geometry evolution under icing conditions. This new method needs to include a maximum of the following features:

- A formal mathematical definition similar to PDE models;
- The possibility to include a global dependency on the interface location;
- A low computational cost proportional to the surface mesh size;
- A strong robustness in concave regions and ice limit regions;
- A realistic potential for 3D extension.

Some of these features were recognized in the equations defining the hyperbolic mesh accretion which is solved with finite difference to grow a mesh layer by layer with respecting orthogonality between layers. Different smoothing methods included in the equations highly increase the robustness and it also applies to 3D mesh generation [25]. Furthermore, the grids generated with the hyperbolic method feature some similarity to how real ice grows on a surface.

Figure 4.1(a) shows a hyperbolic grid generated on an experimental ice shape and the mesh lines highlighted in blue illustrate how the method handles convex and concave regions. This is very similar to how ice would fill a concave region in experiments with enough water collection and time. On Figure 4.1(b), the orthogonal growth of the mesh lines close to the surface is shown, which is another feature of ice growth. As stated before, the ice growth is directed upwind from the geometry surface and the mathematical formulation of the hyperbolic mesh generation equations corresponds to this behaviour.



(a) Filling of successive concave regions

(b) Orthogonal growth

Figure 4.1 Ice-like features of the hyperbolic mesh generation method (coarsened for display)

4.2 Numerical modelling

This section details how the hyperbolic mesh generation equations (see Appendix A) are adapted to ice geometry evolution. Recall the hyperbolic mesh constraints (Eq. (2.4)) [51, 24]:

$$\mathbf{r}_\xi \cdot \mathbf{r}_\eta = 0 \qquad \text{(2.4a revisited)}$$

$$\hat{\mathbf{n}} \cdot (\mathbf{r}_\xi \times \mathbf{r}_\eta) = \Delta V \qquad \text{(2.4b revisited)}$$

Instead of integrating in time, these constraints are first linearized and then solved with finite difference by marching in the η direction, which acts as a time-like integration variable. As the η direction is normal to the geometry, an orthogonal growth is obtained, similar to what is observed in natural ice accretion. Furthermore, in Eq. (2.4b), the volume (area in 2D) is the driving factor for the layer expansion. It suggests that the ice volume accretion map on the surface could directly be used in the equations. The linearized system of equations of the hyperbolic grid generation is thus rewritten as

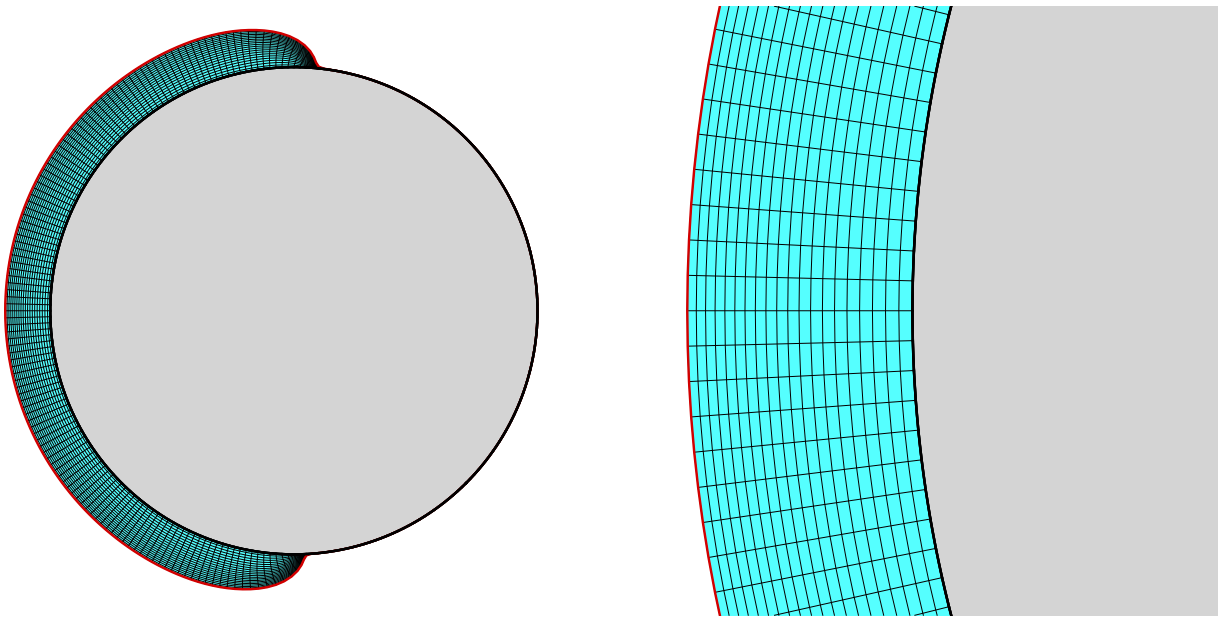
$$A_0 \mathbf{r}_\xi + B_0 \mathbf{r}_\eta = \mathbf{g} \qquad (4.2)$$

where $\mathbf{g}$ is the source term modified with the ice volume (Eq. (4.3)) and A_0 and B_0 are defined in Appendix A. The local ice volume (area in 2D) is computed with the local ice accretion rate obtained in the thermodynamic module such as $\Delta V_{ice} = \frac{\dot{m}_{ice} * \Delta t_{ice}}{\rho_{ice}}$.

$$\mathbf{g} = \begin{pmatrix} 0 \\ \Delta V_{ice} \end{pmatrix} \tag{4.3}$$

The implementation of the hyperbolic ice accretion solver follows that of the hyperbolic mesh generation method described in Appendix A. Each new grid layer is generated by solving a global block-matrix system in which different smoothing mechanisms are included to increase robustness in convex and concave regions as well as to increase the mesh quality. The geometry evolution is thus performed as a mesh generation and results in a volume grid which matches the ice accretion mapping. The final ice geometry then corresponds to the last grid layer that is generated in the process. It is important to note that although the hyperbolic equations are considered stable only for $\Delta V > 0$ [25], the trivial solution with a null volume is still valid in this context and $\Delta V \geq 0$ is stable. Wherever there is no ice accretion, the new vertices are identical to the initial vertices and the new cells have a null volume, without any impact on the final geometry solution. This allows to conserve the clean geometry. An example of solution from the hyperbolic ice accretion solver on a cylinder which half of the surface accumulates a constant ice mass rate is shown in Figure 4.2(a). The gray region is the interior of the initial geometry and the clear blue region represents the volume swept by the grid layers method (shown in black) with the thick red line being the final ice shape. The grid lines are removed from the solutions presented in the rest of this chapter for the sake of clarity.

One main difference over the hyperbolic grid generation solver is the addition of global and local controls over the grid volume, the goal being to evolve the geometry up to the point where the predicted ice volume is reached. As in the algebraic Lagrangian method, the evolving geometry discretization stays the same as the initial surface mesh and it is easy to track the volume swept by the evolving geometry in a local manner by summing up the volume of the new grid cells generated from the surface facets. Whenever the local volume reaches a pre-defined ratio of the predicted ice volume, the volume step by which the new cells are generated starts to reduce according to a user-defined ratio to increase the accuracy with which the predicted volume is obtained. Once it is reached at a local position, the source term (Eq. (4.3)) can be set to 0 and the nodes no longer move even if other sections of the geometry are still evolving. A global control is also necessary to stop the computations after



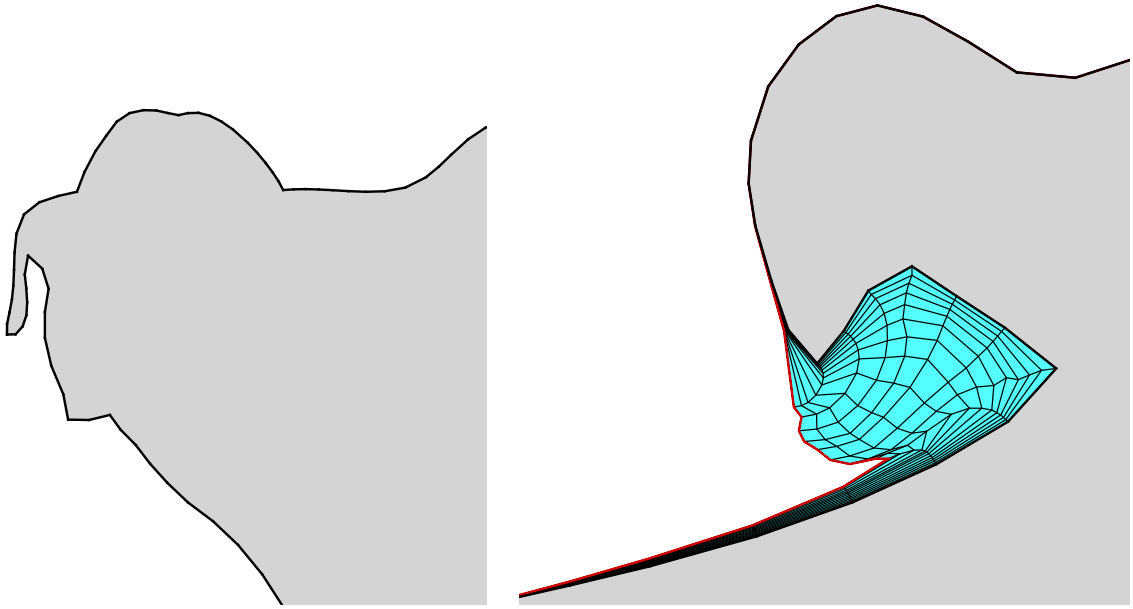
(a) Half iced cylinder (b) Close view of leading edge

Figure 4.2 Examples of ice accretion results from the hyperbolic ice accretion solver

reaching the prescribed global ice volume, as local errors are inevitable in concave regions and at icing limits. Contrary to the Lagrangian method, concave regions are less likely to cause a failure in the simulations depending on the smoothing parameters used and on the severity of the concave region. It is still possible to obtain severe concave regions which even the smoothed hyperbolic ice accretion equations cannot fill with positive volumes only, such as seen in Figure 4.3(a) where the gray region is the geometry interior. Figure 4.3(b) illustrates another concave zone where the method is exploited at its limit, generating small negative volumes internally but still leading to a valid final geometry definition. The gray region is again the interior of the initial geometry, while the clear blue region represents the volume swept by the grid layers generated with the hyperbolic ice accretion method. The correction methods used in the algebraic solver to obtain a valid geometry are thus added to the hyperbolic solver, but these are less likely to be necessary and are added to increase robustness in extreme cases such as the one presented in Figure 4.3(a) which could be closed by one of these methods during the simulation.

A particularly problematic situation can arise when an ice limit is located within a concave region and when the ice height discontinuity is high. In that case, a grid folding might occur as the vertex located at the ice limit stays fixed in place while the neighbouring vertices located in the ice region march away from the surface. It results in a rotation of the neighbouring surface facet around this fixed vertex and in the case where the discontinuity is high, the facet might rotate until it folds onto the clean geometry outside of the ice region. Figure

4.5(a) shows this problem happening in a concave region where a small shadow zone was created, resulting in no water being accumulated in the left part of the concave region while a lot of water is accumulated on the right part. This situation is in fact commonly seen in natural icing and any ice accretion solver must be able to handle it.



(a) Limiting case with a narrow concave region

(b) Near-limiting case for a concave region

Figure 4.3 Limiting cases for the hyperbolic icing method

This problem highlights a drawback seen in explicit interface-tracking methods where there is only local knowledge of the geometry, comparatively to implicit interface-capturing methods where a global knowledge of the geometry is somewhat present. For example, the gradient direction in the level-set equation is always pointing outward from the geometry when the appropriate boundary conditions are respected. Making use of the level-set solver to provide a global knowledge on the evolving geometry to the hyperbolic solver would be too costly in resources as the level-set equation would also need to be solved. However, as the problem arises at the icing limits where the folding occurs on the clean geometry, only a knowledge of the clean geometry is necessary, which could be obtained through the solution of the Eikonal equation.

4.2.1 Link with Eikonal solution

The Eikonal equation (Eq. (3.6)) gives a solution representing the distance to the wall at each point with a unit gradient pointing outward from the simulated geometry. Recall that the "simulated" geometry represents the geometry on which the grid was produced, which

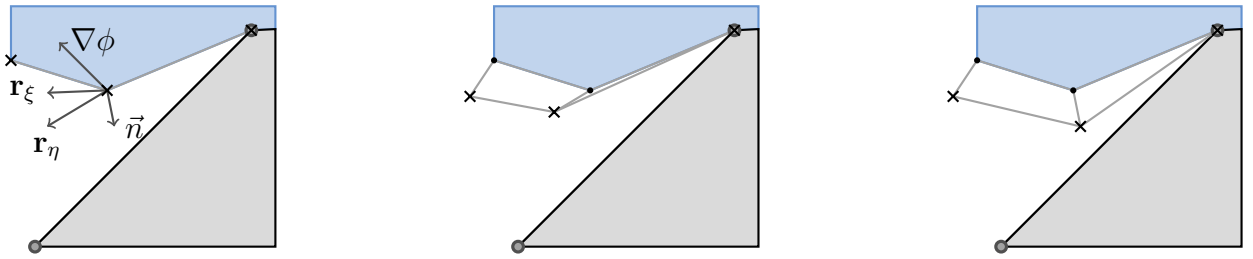
is done at each layer in the multi-layer process, and thus the Eikonal solution is computed once for each of these layers' geometry. The "global geometry knowledge" can be provided by using the Eikonal gradient ($\nabla\phi$) in order to test the hyperbolic ice layer growth direction. In the case where the ice grows in an opposite direction to the Eikonal gradient, it means a folding might occur. Even though only the initial geometry is considered to avoid the folding, using the Eikonal solution has multiple advantages:

1. The boundary condition at the wall is actually forced to ensure the proper definition of the clean geometry at a zero distance;
2. The Eikonal equation does not necessarily represent an additional cost to the simulation as it is used to compute the wall distance required for the turbulence models;
3. As the Eikonal solution does not change during the hyperbolic ice growth, inconsistencies such as symmetric displacement directions in adjacent cells are avoided.

The gradients are obtained at the hyperbolic ice grid vertices by interpolation using a bilinear method, as mentioned in §2.6.2, which is easy to do with the overset grid capability. The source term (Eq. (4.3)) is then modified again to include the interpolated gradient which acts as an external force field pushing the ice growth direction when it deviates from it, leading to Eq. (4.4). Such a deviation could result in the accretion direction being oriented toward the initial geometry, which would lead to an invalid overlap. The scale of the source term depends in part on the sign of the cross product between the Eikonal gradient and the ice growth direction. Furthermore, a relaxation factor (α_ϕ) is added to limit the effect of this external force to an acceptable level. In mathematical terms, this source term allows the ice growth direction to deviate from being orthogonal to the previous sub-layer. It is an original component of this hyperbolic icing method.

$$\mathbf{h} = \begin{pmatrix} \alpha_\phi \operatorname{sign}\left(\nabla\phi \times \mathbf{r}_\eta\right)\left(1 - \dfrac{\mathbf{r}_\eta \cdot \nabla\phi}{\|\mathbf{r}_\eta\| \, \|\nabla\phi\|}\right) \\ \Delta V_{ice} \end{pmatrix} \tag{4.4}$$

Figure 4.4 shows how the source term affects the generation of the sub-layers when the grid normal ($\vec{n}$) is directed toward the simulated geometry, which is known because there is a strong deviation with the Eikonal gradient ($\nabla\phi$) at that location, as seen on the vectors of Figure 4.4(a). The η-direction node derivative ($\mathbf{r}_\eta$) is thus forced to deviate from orthogonality to avoid a future crossing with the geometry, resulting in the new sub-layer displayed in Figure 4.4(b) instead of the one from Figure 4.4(c) where the source term is not active.



(a) Vectors at node location

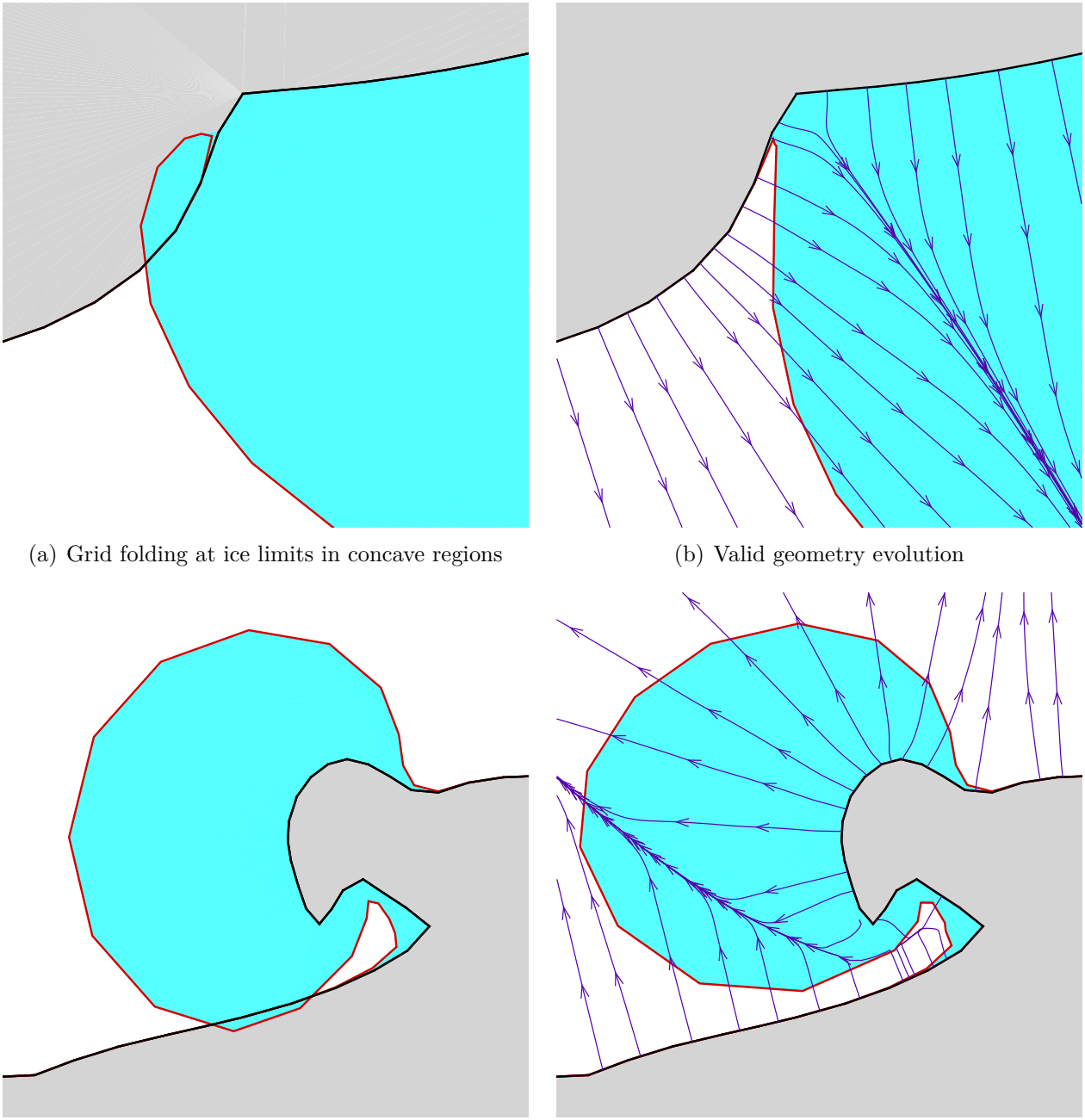
(b) New sub-layer with source term

(c) New sub-layer without source term

Figure 4.4 Local effect of the modified hyperbolic source term

The latter reduces more the angle of the concavity, which is more likely to cause an issue for the next mesh generation process or for the remaining ice accretion itself.

Comparing Figures 4.5(a) and 4.5(b) shows the impact of the Eikonal gradient deviation source term (**h**) in how the hyperbolic ice accretion solver handles ice height discontinuity in concave regions. Without the Eikonal gradient deviation source term (Figure 4.5(a)), a clear overlap of the new geometry over the initial one appears. In Figure 4.5(b), the Eikonal gradient lines are illustrated with arrows and clearly show how the Eikonal gradient can act as a force field preventing the evolving geometry from overlapping the clean geometry by "pushing" against it. The gray region represents the solid interior. A similar comparison can be made between Figures 4.5(c) and 4.5(d) where ice is generated inside a narrow concave region. In this case, water mass is exaggerated to show that the new source term reduces the risk of getting overlaps of the ice accretion with non-neighbouring parts of the geometry.



(a) Grid folding at ice limits in concave regions

(b) Valid geometry evolution

(c) Grid folding at ice limits in concave regions

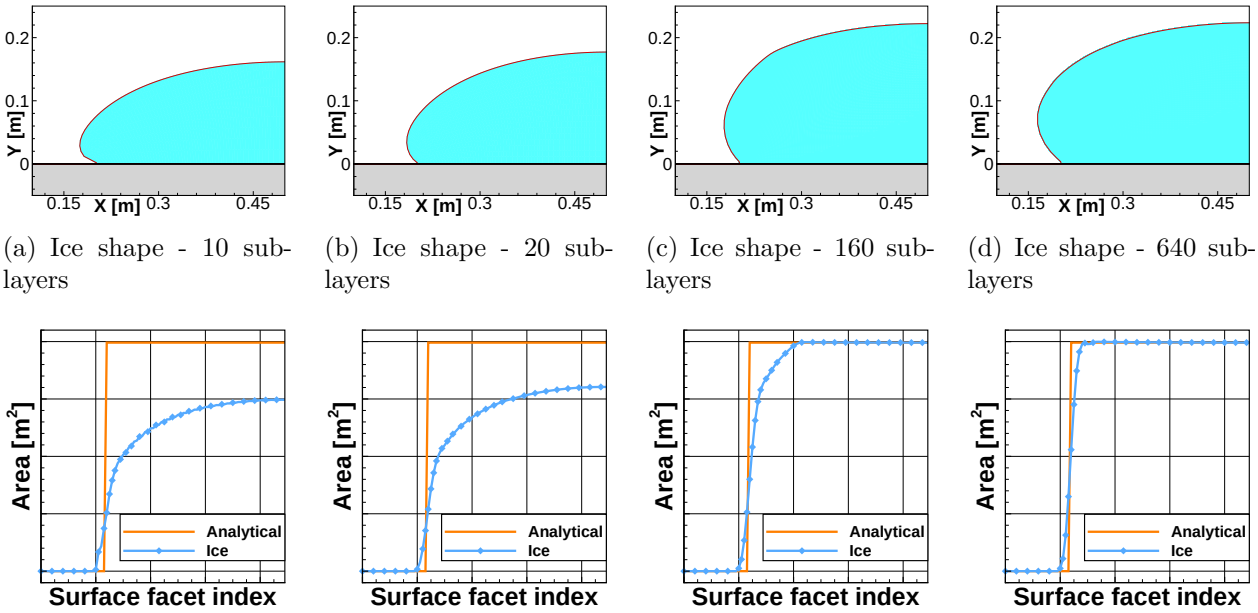
(d) Valid geometry evolution

Figure 4.5 Effect of Eikonal gradient deviation source term in concave region

4.3 Verification of accuracy

Icing on a flat plate is presented to show how the discontinuities in ice height are smoothed and to evaluate the method accuracy order related to the number of sub-layers used. Grid refinement studies are not performed as refining the initial geometry changes the hyperbolic problem and internal controls allow to perform ice accretion to the same accuracy level

regardless of the surface discretization, which does not allow to conclude on a spatial accuracy order.



(a) Ice shape - 10 sub-layers

(b) Ice shape - 20 sub-layers

(c) Ice shape - 160 sub-layers

(d) Ice shape - 640 sub-layers

(e) Area distribution - 10 sub-layers

(f) Area distribution - 20 sub-layers

(g) Area distribution - 160 sub-layers

(h) Area distribution - 640 sub-layers

Figure 4.6 Flat plate canonical icing results

This case is a canonical case where a uniform ice accretion is generated on the centre of a flat plate so to remove any curvature effects of the initial geometry on the results. A ice height of 0.001m is defined in the central section of the flat plate and the hyperbolic ice accretion solver is used to obtain an ice shape. The discontinuity between the ice and the clean region is strong and it is not possible to completely remove the error at this location, especially since it depends on the surface discretization. However, by increasing the number of ice grid sub-layers and leaving the other parameters constant, it is possible to study the order of accuracy of the method when increasing the number of sub-layers, which is akin to a temporal accuracy study. Figures 4.6(a) to 4.6(d) show the ice shapes generated with 10, 20, 160 and 640 sub-layers and Figures 4.6(e) to 4.6(h) show the ice area distribution compared to the predicted ice area distribution. Cases were run by doubling each time the number of sub-layers from 10 to 640 and the global error of each case is shown in Figure 4.7 in relation to the number of sub-layers. The angle between the flat plate and the ice is due to the vertex at the ice limit being fixed in place, resulting in a rotation of the first iced facet.

It can be seen that for such a strong discontinuity, the ice accretion must be divided in much more sub-layers than in real cases where 100 sub-layers were found to be enough. Figure 4.7 confirms this as the convergence in sub-layers stabilizes around a first order accuracy

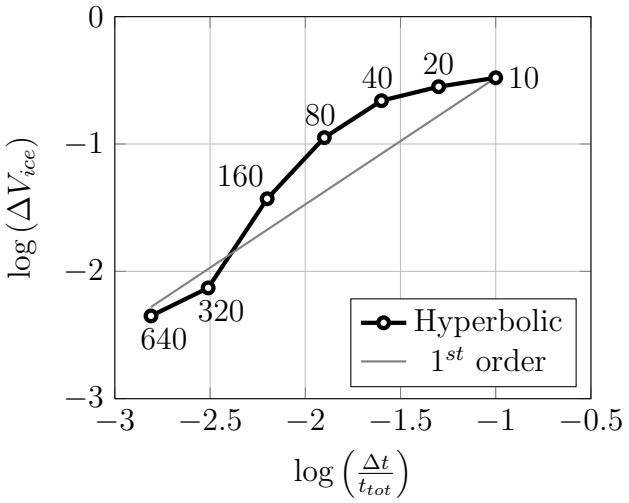


Figure 4.7 Results of the sub-layers order of accuracy study

trend only after more than 300 sub-layers. However, this number of sub-layer remains case dependent. Furthermore, adding more sub-layers might not improve the results as there is an internal control over the local expansion ratio which allows to obtain the local predicted ice area with high accuracy by reducing the size of the sub-layer cells. Regardless of the user-defined number of sub-layers, the final sub-layers are all similar in size once enough layers are used, thus resulting in a stable error level which depends on other user inputs, such as the local tolerance or the minimal sub-layer size.

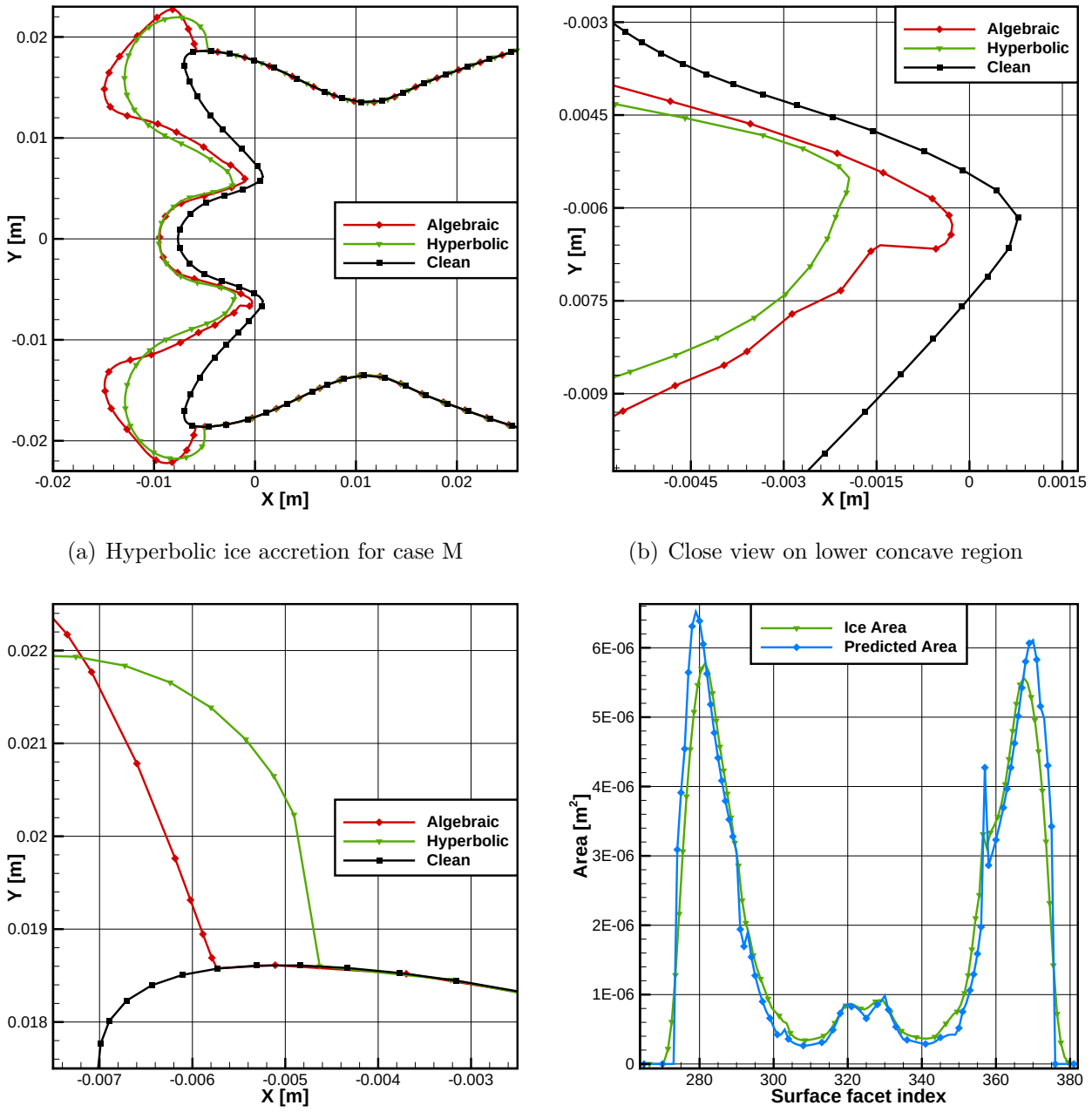
4.4 Icing results

In this section, icing results are presented on the cases studied for the level-set equation in Chapter 3. First, the manufactured case presented in §3.7.1 is studied to show how the hyperbolic ice accretion method handles ice accretion in concave regions and how the icing limits are defined smoothly. Then, the two experimental icing cases studied in §3.7.2 and §3.7.3 are simulated using the hyperbolic ice accretion solver. The global and local errors for these two experimental cases are also presented. The details on the case parameters can be found in Table 3.1 from §3.7.

4.4.1 Manufactured Case

This first case is a purely numerical case whose purpose is to compare the hyperbolic icing solver to the algebraic iterative approach presented in §2.5.1. It was already stated that this manufactured geometry contains three ice-like horns which generate four recirculation zones and four water accumulation points. Using this specially designed geometry allows to

evaluate how a geometry evolution method handles concave regions. Although this point was already tested in the previous canonical cases presented for the hyperbolic icing solver, this manufactured case ice accretion is obtained with the complete icing framework and allows to better assess the method in a natural icing situation.



(a) Hyperbolic ice accretion for case M

(b) Close view on lower concave region

(c) Close view on ice region limit

(d) Hyperbolic local ice area distribution

Figure 4.8 Manufactured results for hyperbolic and algebraic methods

Figures 4.8(a) to 4.8(c) compare the hyperbolic ice accretion method to the algebraic method. Note that although the case is the same as the one presented in §3.7.1, the grid is different and

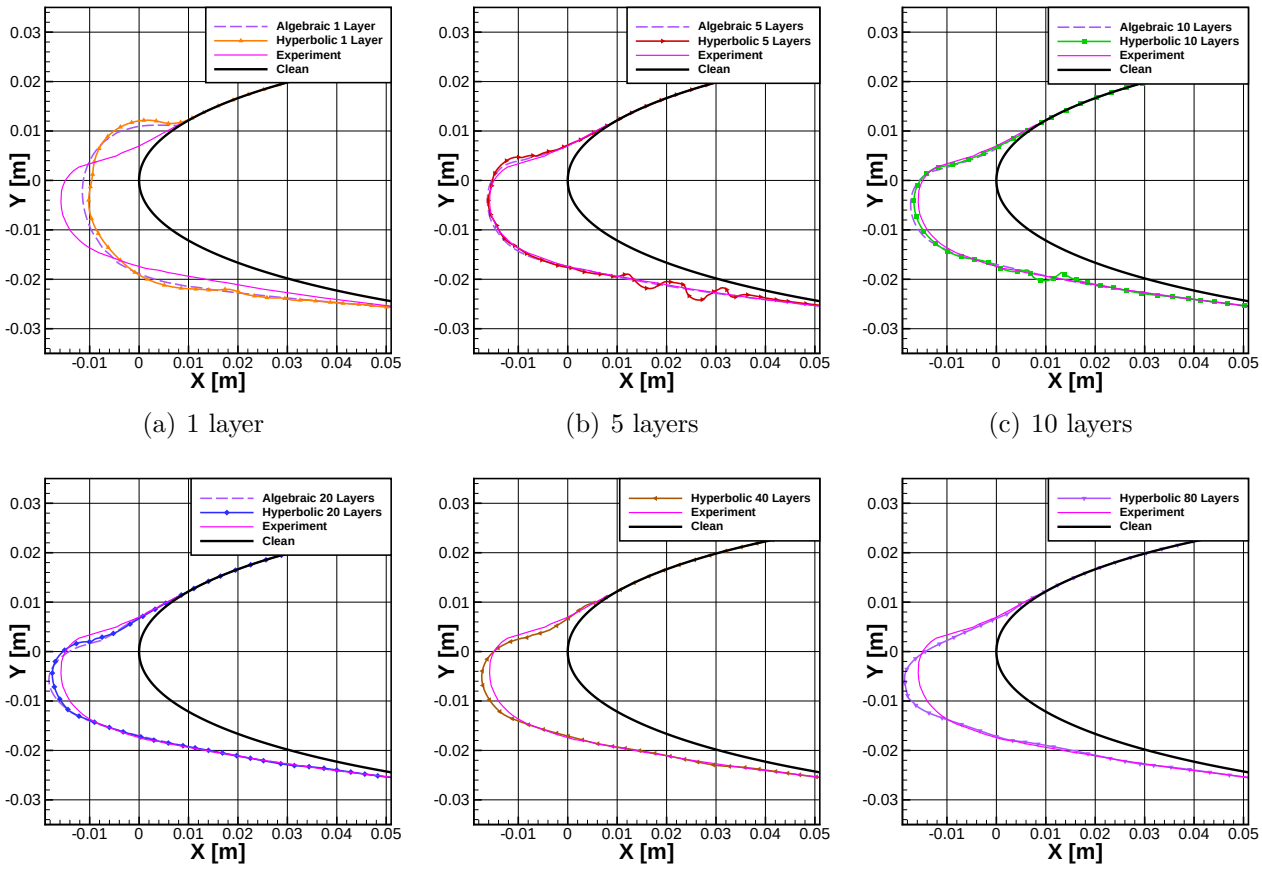
the algebraic results were thus regenerated. The improvement of the correction techniques ensuring the production of valid geometries explains why the algebraic results in these Figures do not include overlaps, contrarily to what is shown in §3.7.1. However, these different correction methods perform geometrical operations which do not conserve the volume, which is why the algebraic method yields a different shape than the hyperbolic ice accretion method. In fact, the global shape obtained with the hyperbolic ice accretion method makes more sense than the shape obtained with the algebraic method, as the ice accumulated on the horns is more curved and the concave regions are being filled more uniformly. In Figure 4.8(b), it can be seen that the hyperbolic result in the concave region resembles more qualitatively to a curve extruded normally to the initial curve, as it would be obtained in a CAD software. In Figure 4.8(c), it can be seen that the ice limit obtained with the hyperbolic ice accretion method is farther along the surface than with the algebraic method. This is due to the different smoothing mechanisms included into the hyperbolic equations, but also to the higher number of sub-layers used in the hyperbolic method. As the hyperbolic ice geometry evolves, the growth direction evolves as well and a rotation of the limiting surface facet is obtained. This rotation is counterbalanced with the Eikonal gradient deviation source term and results in the limiting surface facet to increase in size instead of purely rotating, which can be seen in Figure 4.8(c) with the almost vertical limiting facet of the hyperbolic solution. Finally, Figure 4.8(d) shows the distribution of the local ice area error compared to the predicted ice area as obtained with the thermodynamic model. It can be seen that the local predicted area is not obtained exactly at all locations on the surface. In fact, less ice is accumulated on the tip of the upper and lower horns, but this is compensated in part by the smoother ice limits and a small increase in ice area in the surface facets between these two horns. Globally, a numerical error of 2.13% between the accumulated ice area and the predicted ice area is obtained, which is reasonable for a single layer of ice on such a geometry. For the purpose of comparison, a numerical error of 2.73% is obtained with the algebraic method.

4.4.2 Rime Ice

The first experimental case is a rime ice case which corresponds to "CASE_01" in Trontin *et al.* [100] and which has been presented for the level-set method in §3.7.2. It was run in 1, 5, 10, 20, 40 and 80 layers with the hyperbolic ice accretion method presented in this chapter and the results for each simulations are shown in Figures 4.9(a) to 4.9(f) and compared to experimental data. The results obtained with the hyperbolic method are also compared to results obtained with the algebraic method only up to 20 layers, as in §3.7.2, as the simulations on 40 and 80 layers failed with the algebraic method. The result for 5 layers of ice is very close to the experimental data, but it is not numerically converged in ice layers.

By further increasing the number of layers, it can be seen that the ice shape slowly converges toward a more slender shape than the experimental data. In particular, small bumps of ice are accumulated on the lower section when using 5 and 10 layers, but then disappear with more layers. However, even though these bumps are not part of the converged solution, all simulations with 5 layers and more are in good qualitative agreement with the experimental data.

Table 4.1 presents the percentage of global gain or loss in ice mass for this rime case. Recall that this error is computed by comparing the sums of the accumulated ice volume and of the predicted ice volume at each layer to remove any error coming from re-meshing or spatial discretizations. The error below 0.01 % which can be observed for all simulations with the hyperbolic method shows the high accuracy of the method ensured by the local and global controls.



(a) 1 layer (b) 5 layers (c) 10 layers

(d) 20 layers (e) 40 layers (f) 80 layers

Figure 4.9 Hyperbolic icing solver results for the NACA0012 rime case 1

Table 4.1 Rime case ice mass error

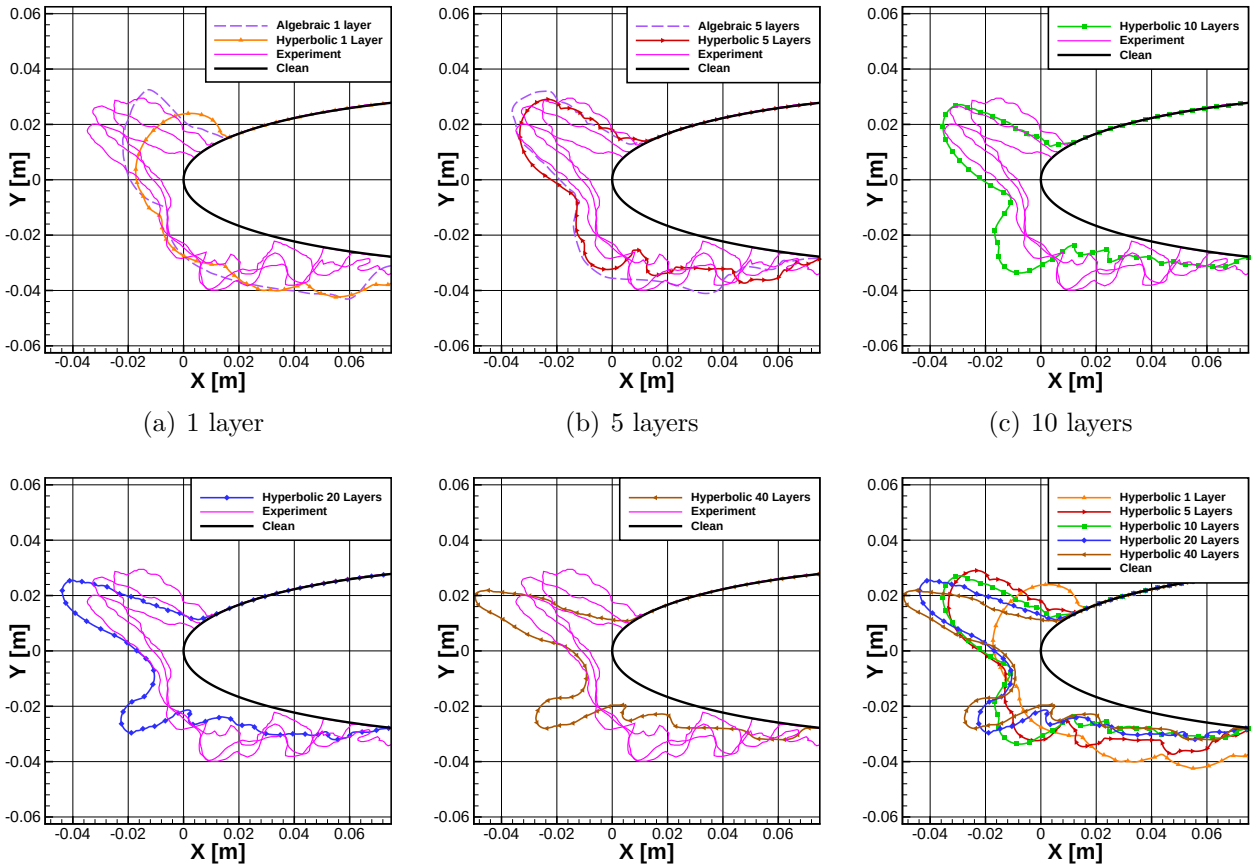
Case	Hyperbolic global error	Algebraic global error
1 Layer	−0.002%	0.98%
5 Layers	−0.002%	0.32%
10 Layers	−0.002%	0.14%
20 layers	−0.002%	−0.16%
40 Layers	−0.003%	-
80 layers	0.006%	-

4.4.3 Glaze Ice

The second experimental case is a glaze ice case which corresponds to "Condition No. 3" in Trontin *et al.* [100] and is presented for the level-set method in §3.7.3. It is run in 1, 5, 10, 20 and 40 layers with the hyperbolic ice accretion method presented in this chapter and the results for these simulations are shown in Figures 4.10(a) to 4.10(e) and compared to experimental data. The results obtained with the hyperbolic method are also compared to results obtained with the algebraic method only up to 5 layers, as in §3.7.3, as the simulations on 10 to 40 layers failed with the algebraic method. As expected, the result for 1 layer is quite different from the experimental data, even though the lower horn region is well defined in ice thickness. The result for 5 layers of ice is very close to the experimental data, but the strong variations seen while increasing the number of layers to 10, 20 or 40 layers indicate that this similarity is a coincidence. By increasing the number of layers, it can be seen that the two main horns of the ice shape are becoming thiner than the horns in the experimental data, but also that their growth direction is moving toward the front.

This change can be seen on Figure 4.10(f) where the different results are shown together. Smaller horns are generated in the lower section of the ice, with qualitative shapes and thicknesses similar to the experimental data in the same region. Overall, the large deviation from the experimental data is not related to the ice accretion method itself, but on the global icing process and on different parameters such as the surface roughness which probably induces the heat transfer to be slightly overestimated in the region around the stagnation point. This results in more water freezing in this region and water flowing less far on the lower and upper sides, yielding thinner upper and lower main horns that are closer to the stagnation point.

Table 4.2 presents the percentage of global gain or loss in ice mass for this glaze case. An error below 0.01 % can be observed for all simulations using the hyperbolic method except for the one with only 1 layer which is not significant, as simulations with only 1 layer of



(a) 1 layer (b) 5 layers (c) 10 layers

(d) 20 layers (e) 40 layers (f) Comparison

Figure 4.10 Hyperbolic icing solver results for the NACA0012 glaze condition 3

ice present high discontinuities in ice accretion, especially for glaze cases. Again, this shows that the hyperbolic ice accretion method is highly accurate in deforming the geometry with the correct amount of ice predicted by the thermodynamic model. It also shows that the deviation of the simulated ice shape with respect to the experimental data is not related to the ice accretion method.

Table 4.2 Glaze case ice mass error

N Layers	Hyperbolic global error	Algebraic global error
1 Layer	−0.068 %	1.19%
5 Layers	−0.007 %	0.28%
10 Layers	−0.006 %	-
20 layers	0.0007%	-
40 Layers	0.0002%	-

4.5 Conclusion

In this chapter, an original ice accretion method is presented based on the hyperbolic mesh generation equations. The hyperbolic ice accretion method generates grid sub-layers to model the geometry evolution due to icing. The ice accretion mapping on the surface is directly used in the equations as a source term controlling the sub-layers generation. Furthermore, local and global controls are added to ensure that the predicted ice volume is reached to a user-defined tolerance with high accuracy, e.g. by reducing the sub-layer growth when approaching a local ice volume target.

One originality resides in the addition of a new source term in the equations in order to introduce a global knowledge on the initial geometry location in the equations. This source term is based on the deviation between the ice accretion growth direction and the gradient of the Eikonal equation obtained in the field surrounding the geometry. The gradient lines can be visualized as a force field which allows the grid sub-layers to deviate from orthogonal growth in order for them to grow away from the clean geometry. The effect of this source term is demonstrated on various canonical cases featuring severe concave regions where turning off the source term results in invalid geometries which overlap the initial geometry.

The accuracy order in sub-layer number of the hyperbolic icing method is verified to be of order 1, but surface discretization effect on the accuracy could not be tested. A manufactured case on a three-horned shape shows how the method handles concave regions in realistic icing conditions and two experimental cases are also tested. In these two cases, although the glaze ice case does not converge with accordance to the experimental data, both show that the ice accretion is representing with high accuracy the ice volume which is predicted to be accumulated. The global errors obtained are all in the order of 0.01 % or below, which is considered conservative in mass.

The hyperbolic ice accretion method contains all elements sought for a robust ice accretion method that are listed at the beginning of this chapter. It is a PDE-style method solved with finite differences which is proportional in computing cost to the surface discretization, thus an efficient method. A global dependency on the initial interface location was added successfully through an original source term, leading to increased robustness in concave regions and ice discontinuities. The method also has a strong potential for 3D extension as it is based on the hyperbolic mesh generation equations which are known to work well in 3D. Overall, it is the most robust method available in NSCODE-ICE, while also featuring the best natural mass conservation properties and a very high efficiency in computing cost.

CHAPTER 5 STOCHASTIC ICE ACCRETION

The icing process is known to be chaotic by nature with supercooled droplets distributed unevenly in a cloud and impacting a solid surface with randomness. This chapter breaks with the deterministic and continuous modelling approaches seen in the simulation community by presenting a heuristic method to model the ice accretion with randomness and probabilities. The method extends the single layer models of Szilder and Lozowski [95], applying the previously developed algorithms to multi-layer simulations. The heuristic numerical developments are presented and compared to the existing model. The impact of the model parameters is then studied. Finally, ice accretion results are compared to experimental data.[1]

5.1 Literature

In the domain of aircraft icing simulations, most of the efforts are done to improve the fidelity and accuracy of the existing modules (aerodynamic flow, droplet trajectories, thermodynamic exchanges), e.g. by using PDEs. For example, recent thermodynamic models were developed to address different aspects of the ice accretion process, such as runback water height (e.g. Shallow Water Icing Model (SWIM) [10] or Extended Messinger [62]). However these models are continuous and do not allow the accretion of small morphologies (e.g. rime ice feathers) nor strong three-dimensional ice morphologies (e.g. scallops forming on swept wings). This limitation comes in part from the front evolution methodology which relies on a continuous front (either explicit or implicit). Another source of limitations is the impossibility to include small imperfections in continuous geometries, but such small discontinuities provide an initial nucleation point for ice crystals to build and generate the mentioned complex ice morphologies.

In comparison, Szilder and Lozowski [95] presented the morphogenetic approach that models the ice accretion with stochasticity and accumulates the ice with building blocks akin to pixels on a Cartesian grid. These blocks represent clusters of liquid droplets that are moving toward the solid geometry using a Lagrangian particle trajectory solver [94]. Upon impact, the fluid elements can freeze or move along the surface of the geometry which is modified continuously with freezing fluid elements. In the morphogenetic model, stochasticity is introduced in two key aspects of the icing simulation process: i) the droplet impingement and ii) the random motion and freezing of fluid elements on the surface [92].

[1] Parts of this chapter are based on reference [12].

First, early results were simulated without a particle trajectory solver and instead relied upon a parametric definition of the impingement on the surface acting as a probability function from which a stochastic impingement was extracted [95]. Later on, with the addition of a complete trajectory solver, the stochasticity was introduced in the seed points selection for the Lagrangian particle trajectory method [94]. Butnarasu *et al.* [22] also developed a version of the morphogenetic approach in FENSAP-ICE in which the droplet trajectories are integrated from an Eulerian droplet solution.

Second, the thermodynamic balance equations are combined locally in the form of a freezing probability that is compared to a pseudo-random number (PRN) to decide whether a fluid element freezes in place or moves. The freezing probability itself is modulated by the distance to the stagnation point or stagnation line with the probability increasing when a fluid element is moving downstream. The water runback movement on the surface is also driven by uniform probabilities compared to pseudo-random numbers. The freezing and movement of fluid elements are performed on a Cartesian grid where the surface corresponds to the cells internal to the initial geometry or frozen that are in contact with the air [92].

The main impact of this method is the natural generation of complex shapes of ice, such as scallops on swept wings (Figure 1.8(d)) or rime feathers on airfoils [92]. In counterparts, the method requires the use a secondary Cartesian grid to accumulate the ice and it increases the difficulty to generate new meshes on the accreted geometry, thus limiting the possibility to perform multi-layer ice accretion simulations. The modified approach from Butnarasu *et al.* [22] presents an algorithm to perform such multi-layer simulations in 2D, but without much detail on how the geometry is extracted at each layer.

To conclude, the heuristic concepts of the morphogenetic are based on physical evidence from experimental data and have shown to yield results qualitatively comparable to experiments, but the implementation details have an important effect on the results [95]. Furthermore, the model is not represented by continuous equations, but by the concept of stochastic accretion of discrete fluid elements in a building block manner.

5.2 Numerical modelling

The non-deterministic ice accretion modelling of the morphogenetic method is unique among the different ice accretion methods available in the literature for aircraft icing. It presents a real potential to model adequately the ice morphologies that deterministic and continuous models are unable to represent. In NSCODE-ICE, a stochastic ice accretion method is implemented following the concepts of the morphogenetic model [95, 92] with significant

modifications to the numerical implementation to improve upon limitations noted in the literature. These modifications reside in three key aspects of the model listed below:

- the secondary grid and front identification;
- the droplets impingement;
- the stochastic freezing process.

5.2.1 Global process

Figure 5.1 illustrates how the stochastic ice accretion is performed. The three key aspects mentioned earlier are strongly linked in the inner loop where the impingement, freezing and front update phases are performed sequentially. The initial front creation depends entirely on the grid and on the hole cutting algorithm from the overset grid preprocessor. The identification of the geometry, which is a hole in the grid layout, allows to build an initial front adequately delimiting the solids. The droplet field solution, the collection efficiency (β) and the freezing fraction (f_{wall}) are interpolated on from field and wall donors at the creation of any Cartesian front cells. Depending on the droplet velocity vector and on the neighbour's status, the cells are classified in different categories which determine the potential for impingement. The actual impingement and freezing processes are then performed and the front is updated each time a fluid element is frozen. More details are given in the following sections.

5.2.2 Cartesian advancing front mesh

A secondary grid made of Cartesian cells is best suited to perform the stochastic ice accretion as the cells have a constant dimension. However, adding a complete Cartesian grid on top of the geometry, as done in the literature [95, 22], consumes too much computing resources since the cells' dimensions need to be small which increases the number of cells with at least half of them being useless due to the overlap with the geometry. To solve this issue, the process detailed in Figure 5.1 contains an original algorithm to provide a suitable grid which differs from a Cartesian grid laid onto the body-conforming mesh used by the other modules.

First, the geometry is identified using an internal constraint Delaunay triangulation grid coming from the overset grid preprocessor of NSCODE-ICE. Then, the front is created around this geometry in an unstructured Cartesian advancing front process. The front cells are defined as having a neighbour in the geometry or in the ice. Around those front cells, two layers of halo cells are created in order to simplify the freezing front updating process. From

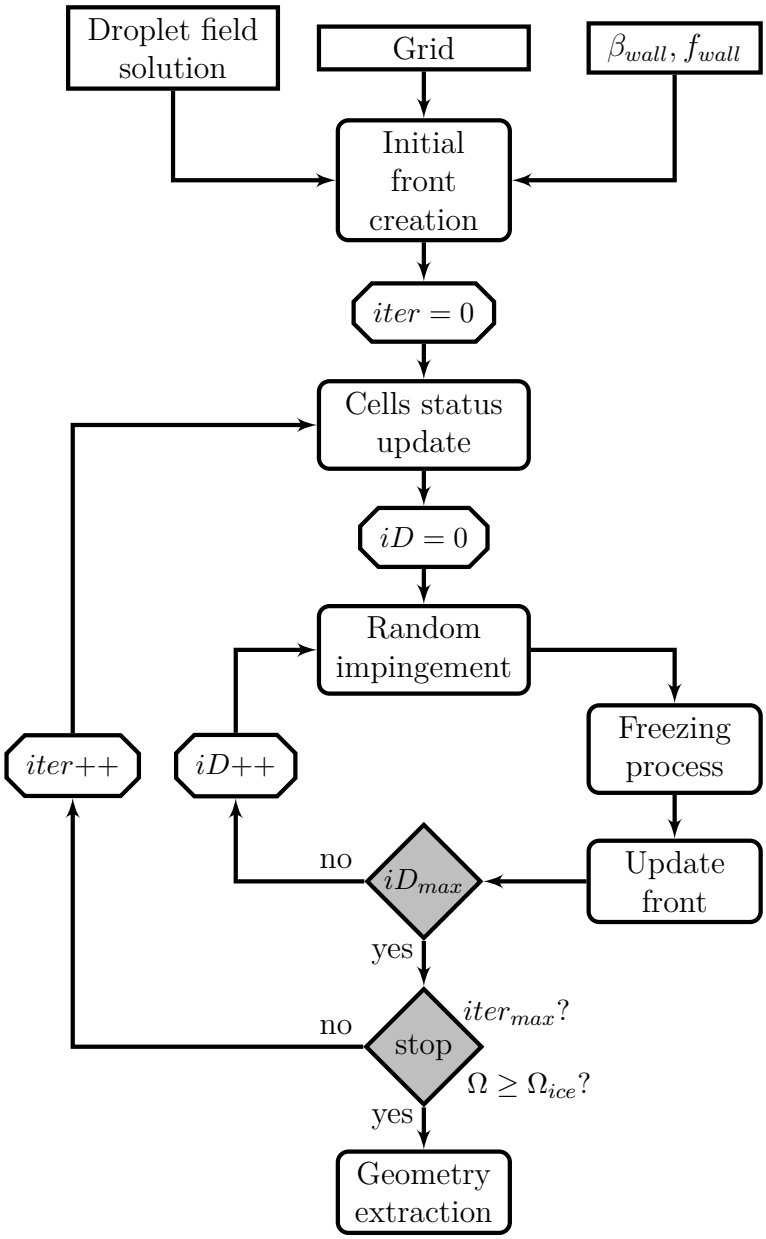
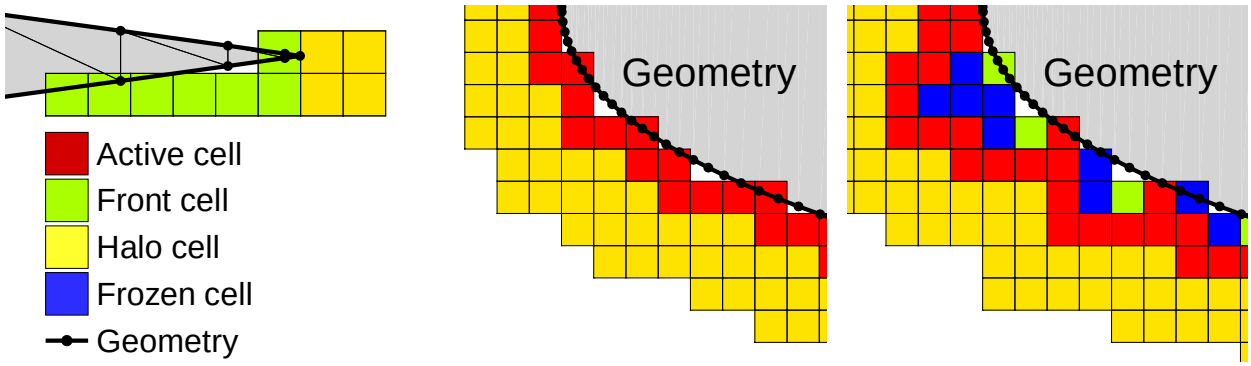


Figure 5.1 Process of the stochastic ice accretion

the front cells, some may be impinged by droplets depending on their position, their neighbour cells and the droplets field and are thus called active cells. Figure 5.2(b) illustrates this front organization while Figure 5.2(c) shows an example of an updated front after a cell freezes. It allows to reduce the computation cost in both memory usage and time requirements. Furthermore, all Cartesian cells are linked to donor nodes from the domain mesh at creation through the overset grid interpolation process, thus obtaining interpolated values of the flow and droplets variables as needed. These are useful in order to compute local impingement characteristics, thermodynamic parameters and water runback behaviour with skin friction.

The interpolation is done with a least square interpolation weighted with the inverse distance between the centre of the Cartesian cell and the donor nodes.



(a) Front creation (b) Initial front (c) Updated front

Figure 5.2 Advancing front organization

5.2.3 Impingement probability

To make use of the Eulerian droplet trajectory solver already implemented in NSCODE-ICE, the surface collection efficiency (Eq. (2.1)) is directly converted into an impingement probability applied on active cells previously identified. Therefore, the pseudo-random impingement is performed by assigning droplets randomly on the active cells instead of integrating their trajectory from the droplet velocity field solution. This impingement process is similar to the original morphogenetic approach where the impingement is obtained from a parametric equation [95], but differs from the more recent studies [92, 22]. One disadvantage is that it is more difficult to consider the growing ice shape in the random impingement process. However, this is solved by interpolating the droplet velocity vector in each new Cartesian front cell and by comparing it with the nearest solid neighbours. The impingement is possible if the vector is directed toward the solid neighbour and the cell becomes an active cell. Furthermore, these status can change during the ice growth process and cells can be deactivated if they become hidden by ice, such as behind a ice horn.

The active cells are classified in categories according to their local impingement probability and the stochastic impingement follows two steps. First, a pseudo-random number determines the category containing the cell that will receive a droplet. Second, another pseudo-random number defines exactly which cell receives the droplet. In each case, a uniform probability distribution is used to generate the pseudo-random numbers, as detailed in §5.2.6.

The number of droplets which are impinged varies in function of a user-defined parameter, which allows to solve multiple impinging droplets at a time. The required number of droplet

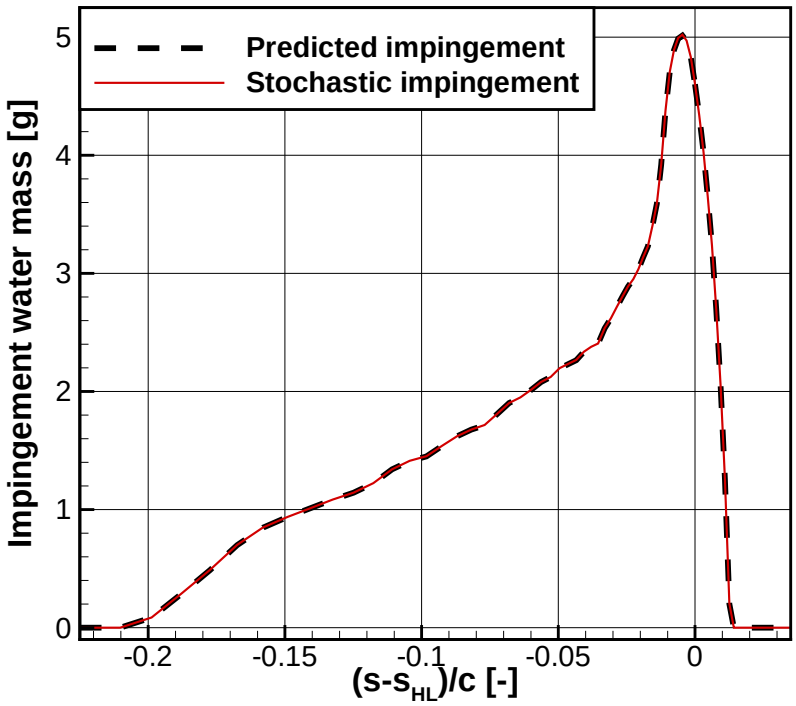


Figure 5.3 Comparison between predicted and stochastic impingement water mass. Figure adapted from Bourgault-Côté and Laurendeau [12]

impingement locations are determined before the freezing process without allowing more than one droplet to impinge a single cell. Furthermore, the water mass impinging rate computed from the collection efficiency is used as a local limiter to prevent the random impingement from accumulating too much water at any surface element. Figure 5.3 shows the capability of the impingement algorithm in matching the predicted impingement water mass. As the two curves are on top of each other, it can be concluded that the impingement mass is conserved even though it is produced pseudo-randomly.

5.2.4 Freezing probability

Once the impingement map is known, the freezing process can start. It depends on a freezing probability (P_{fr}) which is set as the freezing fraction calculated in a standard thermodynamic model on the walls (f_{wall}). The energy and mass balances are computed as usual, but only the freezing fraction is used, not the ice mass accretion rate nor the runback water fluxes. As the thermodynamic model is ran before the stochastic solver is launched, the freezing probability is locally constant during the ice accretion process and the calculation of the energy and mass balances does not hinder the stochastic solver computational efficiency. This aspect of the freezing process is different from the freezing probability used in the standard morphogenetic

model [95] where it depends on the location along the surface and on the number of movement of the fluid element on the surface.

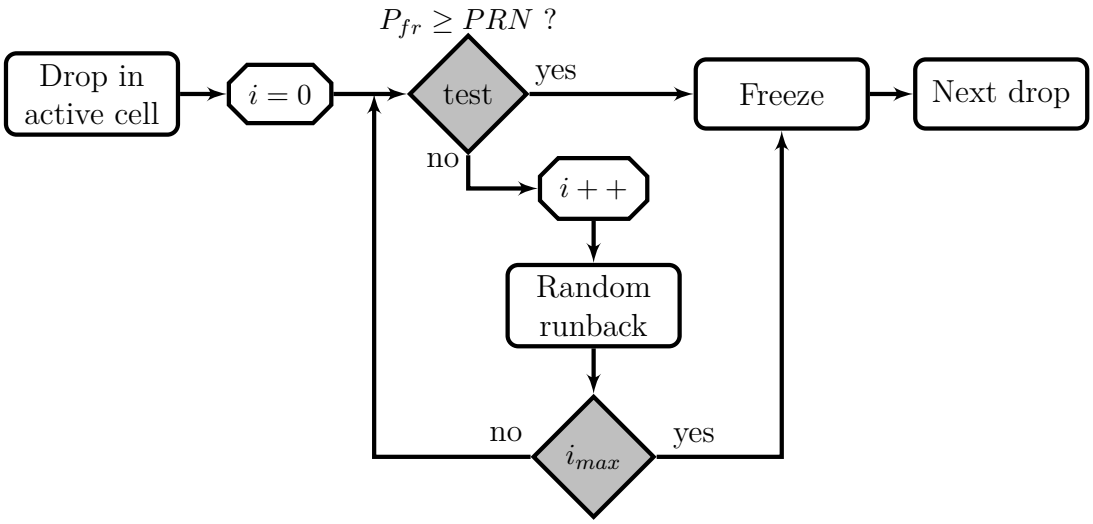


Figure 5.4 Freezing process in the stochastic ice accretion

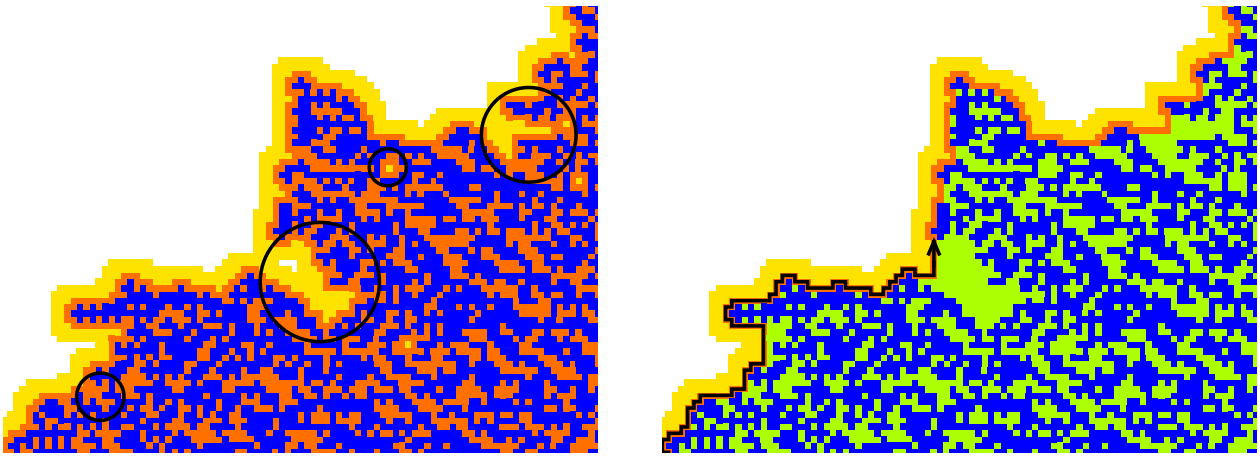
Figure 5.4 illustrates how the freezing process is performed. An impinged cell freezing probability is tested sequentially against a pseudo-random number generated between 0 and 1 with a uniform distribution. If the freezing probability is higher or equal to the pseudo-random number, the fluid element freezes and a proper freezing location must be found by considering the best locations as having more solid neighbours. This process, called cradle search [95], allows to simulate the tendency of water to fill small holes due to surface tension. In the case of rime ice, the freezing fraction is equal to 1 and the cradle search is made into a radius of 1 around the initial location, while this radius is set to 5 in the case of glaze ice. Once a cell freezes, the front is updated around it, as shown in Figure 5.2(c), and another droplet impingement location is selected to test.

In the case where the freezing probability is lower than the pseudo-random number, the fluid element is considered as runback water and flows to another random location from the impinged cell's eligible neighbours. The eligibility of a neighbour depends on its current status, as it must be a front cell, and on the air shear stress, as it was demonstrated to drive the water runback direction [112]. A user defined parameter limits the maximum movement taken each time this situation occurs. Once the defined number of steps is completed, the last location replaces the initial impingement location in the list and is tested again until it freezes. A maximum number of iterations is defined and when this limit is achieved, the water is considered to shed from the airfoil, thus the limit must be high enough so any runback water which would shed must do so after having passed a large portion of the airfoil. When all impinging water freezes or sheds, another round of random impingement starts

and is followed by the freezing process until the global impingement water mass achieves the predicted total mass.

5.2.5 Ice front extraction

In order to use the stochastic ice accretion method in a multi-layer framework, the ice contour must be extracted after each layer and converted into a B-spline. However, this conversion requires an ordered set of data, which is not easily done with the advancing front grid which is generated in an unstructured manner. Furthermore, the stochastic solution itself contains problematic features identified by black circles in Figure 5.5(a). Holes in the grid are seen as white spots surrounded by halo cells (yellow) and highly concave regions are identified by large regions of halo cells and front cells (orange) . The ice front area also contains small ice features similar to dendrites that are unwanted in the data extraction as deemed too small to be represented in the volume mesh.



(a) Raw solution with holes, small dendrites and concave regions

(b) Cleaned solution with single ordered ice front

Figure 5.5 Stochastic ice solution before and after the ice front extraction

To remove these problematic features and ensure a valid data extraction for the B-spline approximation process, a few heuristic rules are set and applied in the following order.

1. Fill the holes with halo cells;
2. Convert halo cells too far from the boundary of the mesh into hidden cells (green);
3. Convert front cells which are not on the interface into hidden cells.

The resulting solution features an ordered list of front cells that can then be used to generate the data required to approximate a B-spline onto. Figure 5.5(b) shows the treated solution

with the new cell status and also the ordered extracted data as a black line with an arrow showing the direction of the extraction.

It could be though that mass conservation would be an issue following this solution cleaning process which includes non-iced cells into the final solution, however, including these cells allows the solver to produce a realistic ice density without having to force it. As the fluid elements that freeze are considered to have the density of water, it accounts for the holes left in the ice by the cleaning phase and that similar to bubbles of air which would be trapped in the freezing process. It is thus possible to obtain different densities of ice depending on the simulations parameters and icing conditions.

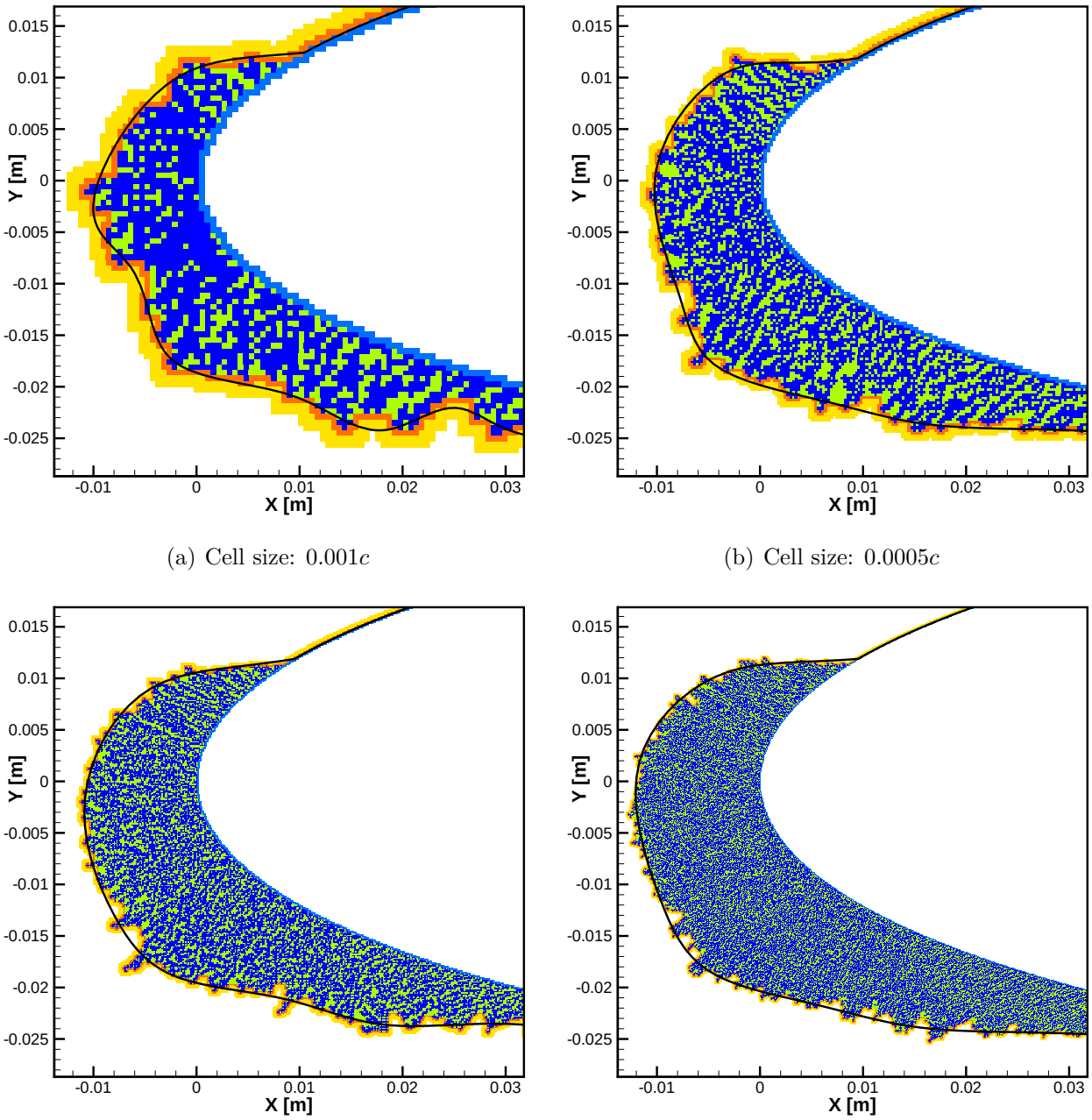
5.2.6 Pseudo-random number generator

In this work, the Mersenne Twister MT19937 pseudo-random number generator from the C++ <random> library is selected as it is recognized in the programming community to give more reliable results than the C <rand> library. Furthermore, the Mersenne Twister engine was developed for scientific applications to yield better quality pseudo-random distributions and has become the default generator in statistical simulations such as Monte-Carlo simulations [41]. It uses the computer clock as a seeding point for the pseudo-random number distribution in order to obtain different distributions at each simulation, but it is also possible to define a constant seed, such as 0, to obtain reproducible pseudo-random distributions.

5.3 Cell size effect

In this section, the effect of the Cartesian cell size is studied on the same rime case presented in §3.7.2 As the solver is non-deterministic, multiple simulations for all cases are run and the results shown are representative of the average between solutions, which cannot directly be established for a geometry. Furthermore, only single-layer results are presented since using multiple layers would introduce error sources in the comparison. The cell size is varied from 0.001 to 6.25e−5 chord length (c). Figures 5.6(a) to 5.6(d) show the results of decreasing the cell size on the stochastic field solution and the resulting B-spline solution. The results obtained by deterministic methods can be studied quantitatively with respect to the ice volume accumulated because the ice density is fixed. However it is not the case for the stochastic ice accretion method. Therefore, the focus of the quantitative analysis is on the ice density instead.

For each size, ten runs are performed to evaluate an average ice density and its standard deviation which are shown in Table 5.1. It can be seen that decreasing the Cartesian cell size



(a) Cell size: $0.001c$

(b) Cell size: $0.0005c$

(c) Cell size: $0.00025c$

(d) Cell size: $0.000125c$

Figure 5.6 Cell size effect on the stochastic ice accretion

allows to converge in terms of density. Figure 5.7 shows that the average ice density converges with second-order accuracy with respect to the stochastic cells size (Δs_{cell}). The converged reference value ρ^*_{ice} used to compute the error in Figure 5.7 is obtained by extrapolating the results with a second-order relation and yields a value of around 631.2kg/m^3. As the expected value would be 880kg/m^3 for rime ice density, a relative difference of 28% in density is obtained for this case. This could be improved by varying the different numerical parameters

of the stochastic ice accretion model, such as the cradle search radius or the impingement process, but it would require extensive parameter studies on multiple cases, which exceed the scope of this work. It must also be considered that the number of ice layers has an impact on the final density, as it can be seen in §5.4.2, whereas for the present simulations only one layer of ice is used.

Table 5.1 Results of the stochastic cell size effect on density

Size	$[c]$	0.001	0.0005	0.00025	0.000125	6.25e−5
Average density	[kg/m^3]	807.6	709.4	648.1	629.3	630.7
Standard deviation	[kg/m^3]	10.6	8.6	3.8	1.7	0.9

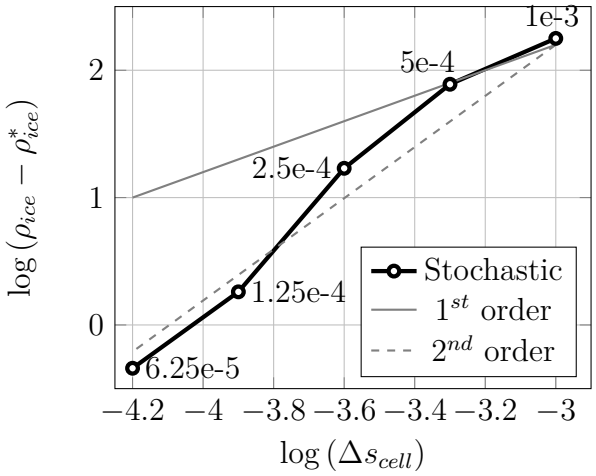


Figure 5.7 Results of the stochastic cell size convergence study

5.4 Icing results

Two experimental icing cases are studied with the stochastic ice accretion solver. These two cases are limited to rime ice conditions as the runback water mechanisms developed in this solver are lacking in accuracy. The first case is the rime case 33 from NASA's technical report on icing on a NACA0012 airfoil [108] and its parameters can be found in Table 5.2. The second case is the rime case named "CASE_01" in Trontin *et al.* [100] which has already been presented in §3.7.2 and §4.4.2. The parameters for this case can be found in Table 3.1. For these two cases, the chord length is 0.5334m and the stochastic cell size is kept constant at $0.00025c$.

Table 5.2 Stochastic icing cases parameters

Case		**Rime Case 33**
AoA	[°]	4.0
Mach	[−]	0.301
Temperature	[K]	242.50
Pressure	[kPa]	92.060
LWC	[g/m^3]	1.05
MVD	[µm]	20
Icing Time	[s]	372
Roughness	[µm]	299.0

5.4.1 Ice shape averaging

The stochastic nature of the method produces different results at each run even with the same parameters and this makes it more complicated to compare results to experimental data or to other methods. To do so, the simulations are performed multiple times and the final ice shapes can be averaged into a mean geometry. This is made possible by the use of the B-splines, as the final curves can be post-processed to be discretized very finely and uniformly in terms of the curve parameter. The discretized curves are then averaged, with each data point corresponding to the same parameter value. As the updated B-splines are built by merging the ice curves into the previous ones, the section without ice corresponds perfectly to the initial airfoil geometry and the averaging process also conserve this geometry.

5.4.2 Rime Ice: Case 33

The first experimental case is a well-known rime ice case on a NACA0012 airfoil presented in Wright *et al.* [108]. It is run in 1, 2, 3, 4, 5, 10 and 20 layers to study the impact of the number of layers on the results with a focus on the evolution of the solutions with a small number of layers. Each case is simulated in five different runs to study qualitatively the variability of the global ice shape due to the stochastic behaviour. Figures 5.9(a) to 5.9(f) show the results obtained for all these runs except for the 1-layer simulations which were removed for the sake of clarity as they are not considered interesting.

Table 5.3 Stochastic density results for rime case 33

Number of layers		1	2	3	4	5	10	20
Average density	[kg/m^3]	698.3	741.3	762.2	782.3	799.5	843.2	866.7
Standard deviation	[kg/m^3]	4.54	25.6	33.9	27.8	28.4	27.4	29.7

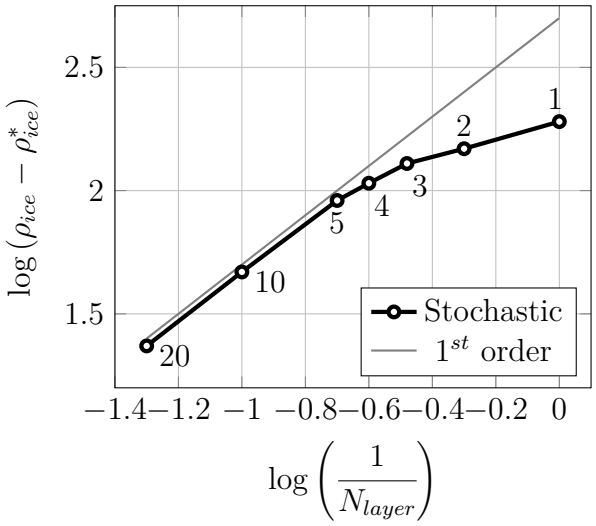
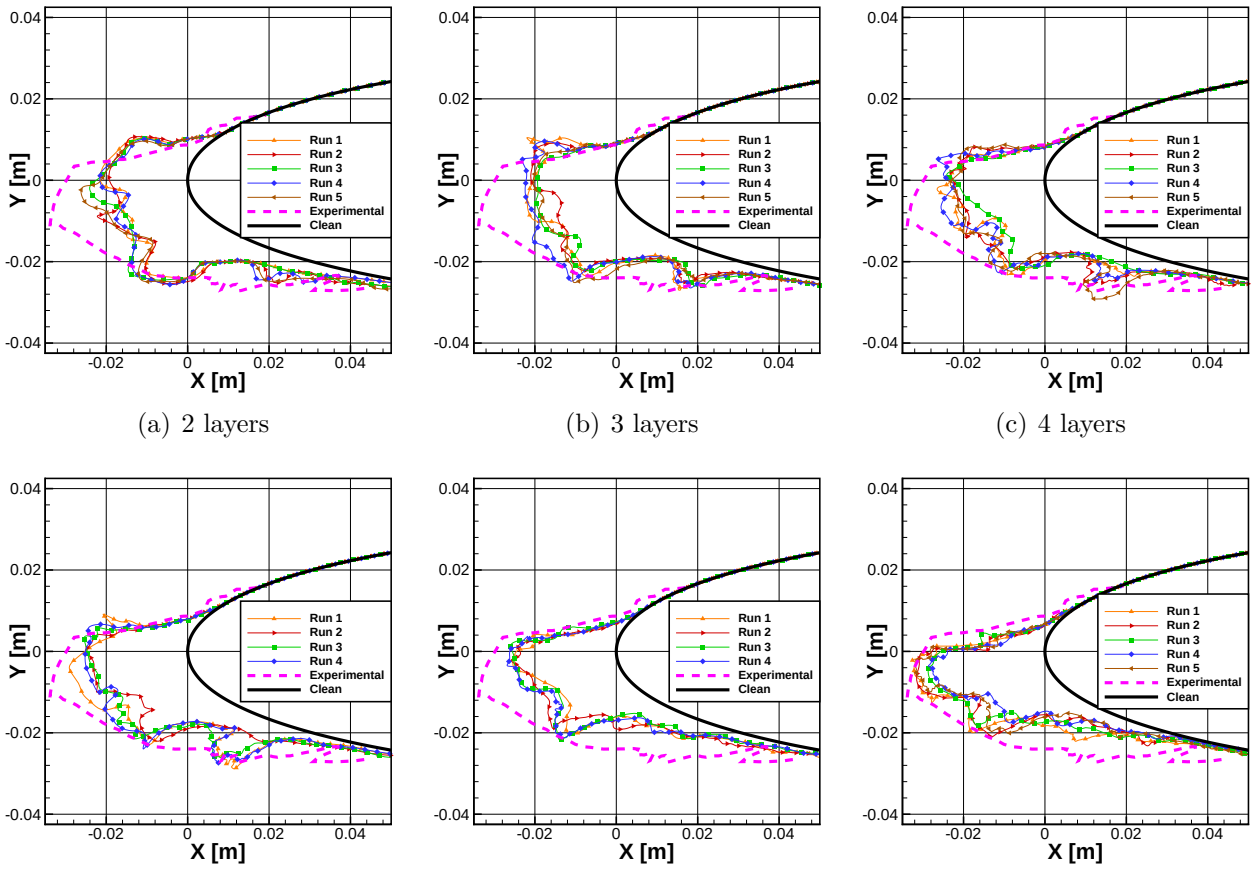


Figure 5.8 Results of the stochastic icing convergence in layers study for rime case 33



(a) 2 layers (b) 3 layers (c) 4 layers

(d) 5 layers (e) 10 layers (f) 20 layers

Figure 5.9 Stochastic icing solver results for the NACA0012 rime case 33

Using only one layer of ice yields a very low variability in the results, even though it is not

shown here, whereas using multiple layers of ice results in a seemingly constant variability level around a mean geometry solution. This variability affects the roughness of the frontal section of the ice more than the width and limits of the ice shapes. As seen with deterministic methods, the ice thickness increases and the ice width decreases when more layers are used, resulting in a more slender ice shape which fits better the experimental data in this case. This trend can better be seen on Figure 5.10 where the averaged curves obtained with the method described in §5.4.1 are shown for each number of layers.

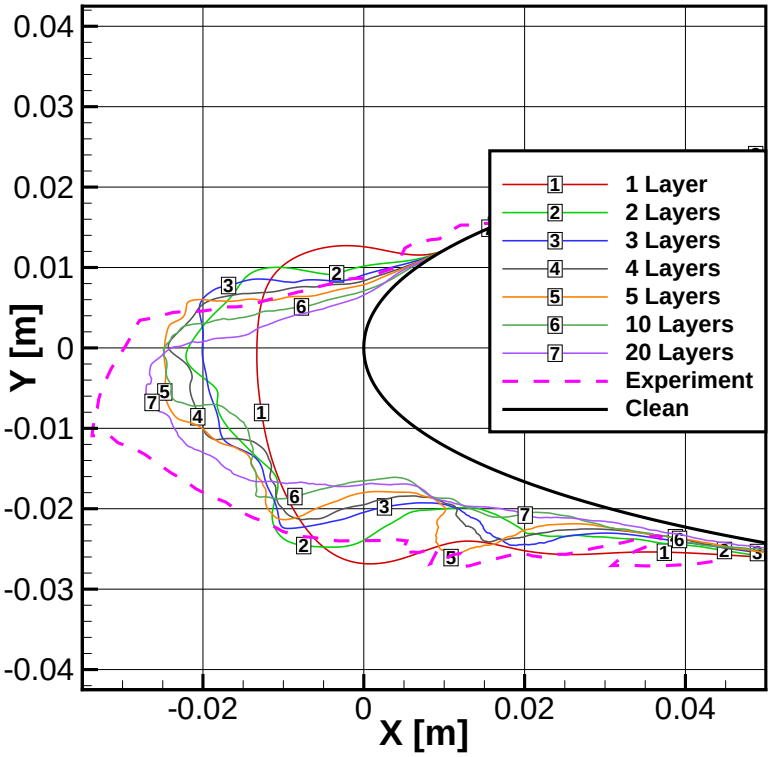
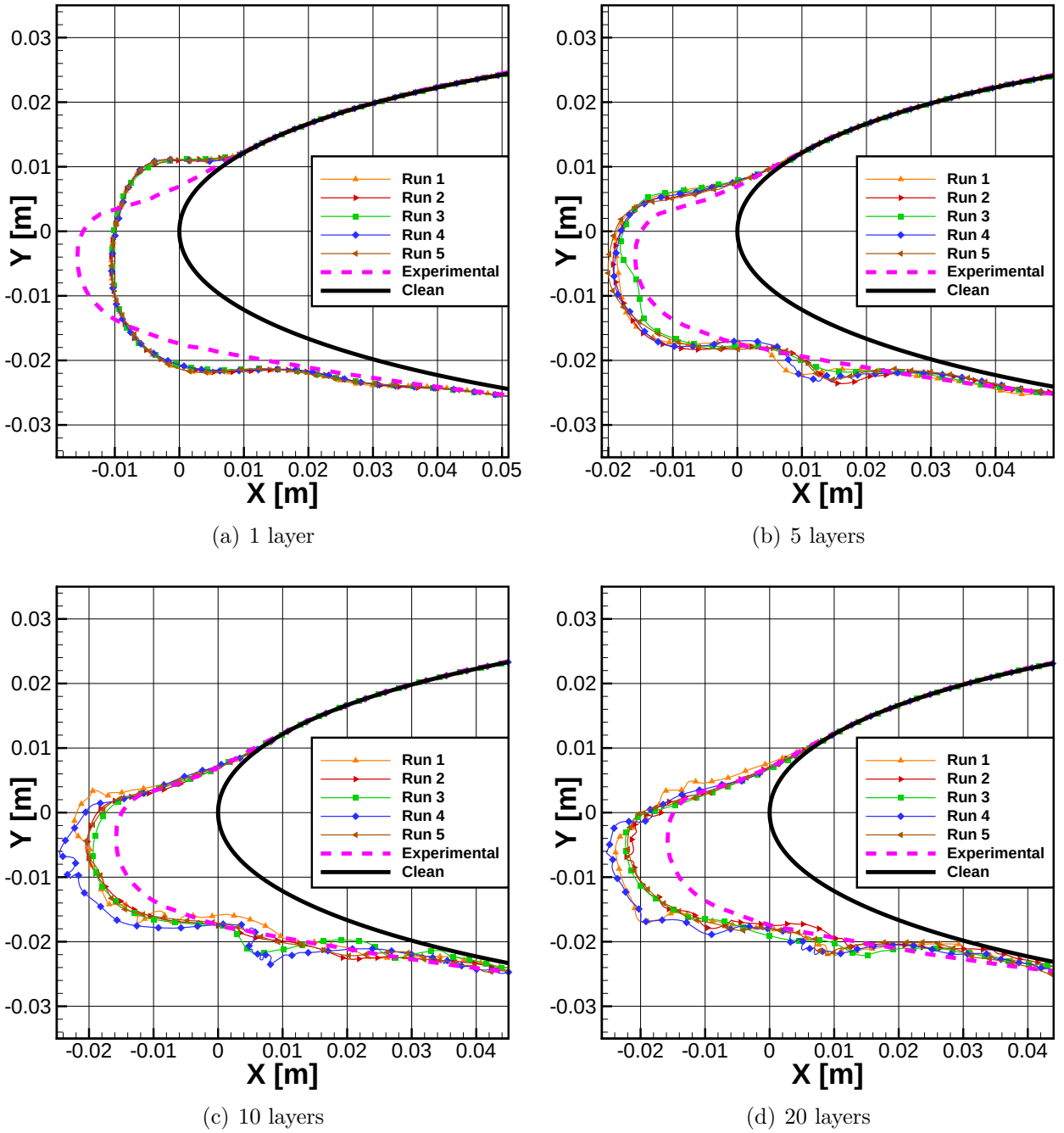


Figure 5.10 Comparison of the averaged stochastic ice shapes for case 33

Globally, the mean ice shape is converging toward the experimental ice shape in a qualitative way. Furthermore, the ice density obtained by averaging the density of all layers for each case is clearly converging toward the typical value of 880kg/m^3, as seen in Table 5.3. This convergence is obtained with a first-order accuracy with respect to the number of layers, as seen on Figure 5.8. The converged reference value (ρ^*_{ice}) used to compute the error is obtained by extrapolating with a first-order relation, yielding a density of 890.2kg/m^3 and a relative error of 1.15% to the expected value. The first-order accuracy convergence rate contrasts with the results obtained in §5.3 where a second-order accuracy convergence is obtained with respect to the cell size. Another difference is the standard deviation which stays roughly constant in Table 5.3 when the number of layers increases, whereas it decreases significantly in Table 5.1 when the cell size decreases.

5.4.3 Rime Ice: Case 1



(a) 1 layer

(b) 5 layers

(c) 10 layers

(d) 20 layers

Figure 5.11 Stochastic icing solver results for the NACA0012 rime case 1

The second experimental case is the rime ice case named "CASE_01" in Trontin *et al.* [100] and is presented for the level-set method in §3.7.2 and for the hyperbolic method in §4.4.2. It is simulated in 1, 5, 10 and 20 layers in five runs each and the resulting ice shapes are shown in Figures 5.11(a) to 5.11(d). As for the previous case, the results variability is negligible

for the 1 layer simulations, whereas it is similar for the other simulations. The ice shapes also become slender when adding more layers, which can be considered a behaviour of the global icing process as it appears regardless of the ice accretion method used. Figure 5.12 shows how the averaged curves are converging toward a result, even though the shapes are still changing significantly between the 10- and 20-layers results. Furthermore, the results are qualitatively in agreement with the experimental data, as the ice shapes for the 20 layers simulation are very similar to the experimental data, even though the numerical ice shape is thicker. In fact, it can be seen on Figure 5.11(d) that the upper and lower sections of the ice are aligned with the corresponding sections in the experimental data. However, the ice density results for this case, listed in Table 5.4, cannot be considered to be converging. Although the values are increasing when more layers are used, as in the previous case, the standard deviation is increasing faster, which indicates that more runs would be required to increase the statistical confidence in the results. Therefore, these results are qualitatively similar to the experiments, but no conclusion can be drawn from the quantitative analysis.

Table 5.4 Stochastic density results for rime case 1

Number of layers		1	5	10	20
Average density	[kg/m^3]	604.8	628.4	640.1	684.8
Standard deviation	[kg/m^3]	2.7	18.1	22.9	86.0

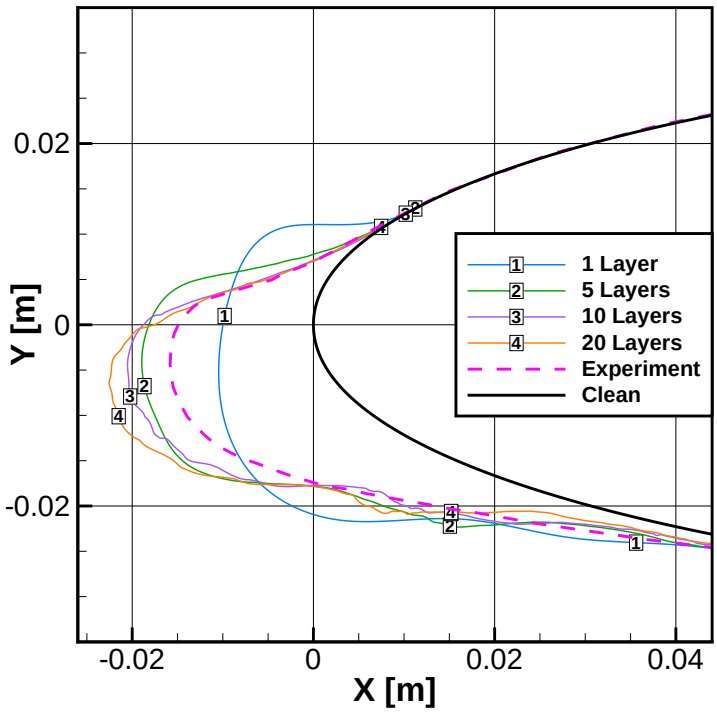


Figure 5.12 Comparison of the averaged stochastic ice shapes for case 1

5.5 Conclusion

This chapter presented a non-deterministic ice accretion method based on the morphogenetic model [95] and adapted to the technologies available in NSCODE-ICE. The stochastic ice accretion method is different with respect to the other methods presented in this study, as it is based on heuristic mechanisms and concepts. Stochasticity is introduced in multiple processes in this method: i) the droplets impingement process, ii) the water elements freezing process and iii) the runback water elements movement process. The impingement process is driven by a local surface impingement probability distribution which is converted directly from the collection efficiency from the Eulerian droplet solver. It is used to avoid the need to track particles which would be necessary in a Lagrangian framework. The freezing mechanism is closely related to the runback water movement process, as both are called in a loop after a water element is impinged on the surface. The freezing fraction given by the thermodynamic model is converted in a freezing probability which determines if a water element freezes at its current location. In the case where it does not freeze, the water element is moved on the surface with some randomness and the cycle continues.

The entire stochastic icing process is performed on an advancing front Cartesian grid which cells are added on demand. It allows to reduce the requirements in computing resources compared to the use of a complete Cartesian lattice where useless cells would be included inside the initial geometry, for example. The ice shapes for each layer are extracted by "filling" the holes and severe concave regions to extract only the outermost layer of front cells before sending the data to the B-spline approximation method. These two aspects of the method, i.e. the grid and the geometry extraction, allow to perform multi-layer simulations with the stochastic ice accretion method, which is not done in the literature. This allowed to observe that a convergence in layers can be reached with a non-deterministic method for the average ice density, but also qualitatively on the resulting ice shapes.

Two experimental rime ice cases were tested in this chapter. For each case, multiple runs were simulated to obtain statistics on the ice density which are used for quantitative analysis. The results are qualitatively in good agreement with the experimental data, but only one of them converges toward the expected ice density within the number of layers simulated. A grid convergence study is also performed with single layer simulations, leading to an observed second-order accuracy in space for the average density.

Overall, original concepts and techniques were included to the stochastic ice accretion method in NSCODE-ICE to obtain multi-layer simulations. The convergence studies show that convergence can be obtained on averaged quantitative values, but more studies are required in

this aspect with simulations using more ice layers. Among the different stochastic processes, the runback water element movement is not working properly, with water elements not moving far enough from their impingement points to represent accurately the runback water film predicted by the thermodynamic model. This is the main reason why glaze ice cases were not presented in this chapter. Therefore, future efforts would be required in this aspect to improve the stochastic ice accretion method.

CHAPTER 6 METHODS COMPARISON STUDY

The previous chapters presented methods to evolve the ice geometry under icing conditions. In this chapter, the three deterministic methods are compared on two different cases to assess their advantages and disadvantages. Quantitative metrics and qualitative criteria are introduced formally and then the results obtained on the experimental cases are shown and compared.[1]

6.1 Background

As it was already mentioned early in this thesis, the geometry evolution module receives less attention than others in the icing literature. Most of the studies specifically dedicated to geometry evolution are considering ice mass conservation only [96] or quantitative criteria to characterize ice accretion for one particular geometry evolution method [89]. By using a quantitative metric such as the sectional global ice area or local ice area distribution on an airfoil, a geometry evolution process can be improved with global and local mass conservation controls to ensure a better quality of the resulting ice shapes. Yet, such quantitative criteria are difficult to define [83] and no ideal metric to characterize ice shapes exists for now to the best of the author's knowledge. Most commonly, global features are extracted from the final solution, such as the ice horns length, angle and position, to assess the quality of the results. The final volume (or area in 2D) can also be compared to the volume of experimental shapes [100], but the large variation of the experimental data along the span of wings and the even larger incertitude between experiments, as shown in Figure 1.4, makes this comparison less representative. This is even accentuated by the fact that multiple legacy experimental data were extracted by drawing the ice shapes sections by hand.

With the technological improvements such as laser scanning of ice shapes [77], it is expected that high quality local and global quantitative comparisons will be possible in the near future. However, the complexity of ice shapes is such that a valid comparison to experimental data cannot be done through a single parameter. As it is shown in the results presented by Son *et al.* [89], a very good qualitative agreement between a numerical result and an experimental shape can still result in a major difference within a quantitative metric such as ice sectional area. In fact, Wright [109] suggests that quantitative measures need to represent the characteristics considered as the most important for the icing simulation objective. For example, aerodynamic performance degradation might be more influenced by the ice horns

[1] Parts of this chapter are based on reference [14].

length than ice coverage, whereas ice coverage might be more significant for the design of de-icing devices. Thus, a thorough approach to evaluate ice shapes and compare them to other results is to consider various criteria representing different aspects of the ice shapes.

In the literature, although the most commonly used metrics for sectional ice quantitative evaluations are the ice area, ice horn length and ice horn angle [109, 89, 100], the way to compute these metrics vary. Wright [109] used the integration of the ice thickness distribution over the curvilinear distance along the clean airfoil to approximate the ice area, whereas Son *et al.* [89] uses the Green theorem on the ice section to compute the real sectional ice area. As for the horn metrics, Son *et al.* [89] computes them in polar coordinates from an origin point set arbitrarily at 25% of the chord. However, as the measures obtained with this approach depends on the origin location, it is limited in significance and cannot be used as a universal criterion [89]. This is different from the method used by Trontin *et al.* [100] where the intersection with orthogonal lines from the initial geometry is used to find the ice horn length and angle, resulting in a more reliable measure. Wright [109] compared a variant of this orthogonal method to a minimum distance approach, finding that the latter is more accurate in general, but that a very small ice thickness requires the former method to keep the same accuracy level. That is due to the discretization errors between the clean and iced surface meshes. These different methods to compute the ice horn length can be extended to compute the global ice thickness distribution. It represents another metric that can be compared, even though such comparison is in part subjective because it is not done through a scalar value [89]. The positions of the minimum and maximum thickness can be of importance in the leading edge region, e.g. for the design of de-icing systems, and the ice coverage can also be found from this distribution.

The present study highlights the main differences between the ice accretion methods by using quantitative and qualitative criteria. The study strictly compares the impact of the geometry evolution method on the results, with the remaining of the modules and parameters remaining constant. In this aspect, it is different from similar studies which have already been referred to in this section, in particular for the extensive validation of codes such as LEWICE 2.0 [109] and IGLOO2D [100]. This is made possible by the wide variety of approaches available in NSCODE-ICE and represents an original study in the icing literature. The stochastic ice accretion method is not included in this study, as it has not reached the same technological readiness level as the other approaches. Furthermore, it cannot be compared with the same metrics due to the variable density obtained with the stochasticity. The following sections describe all the metrics and criteria which are used to evaluate the ice shapes in this chapter.

6.2 Quantitative metrics

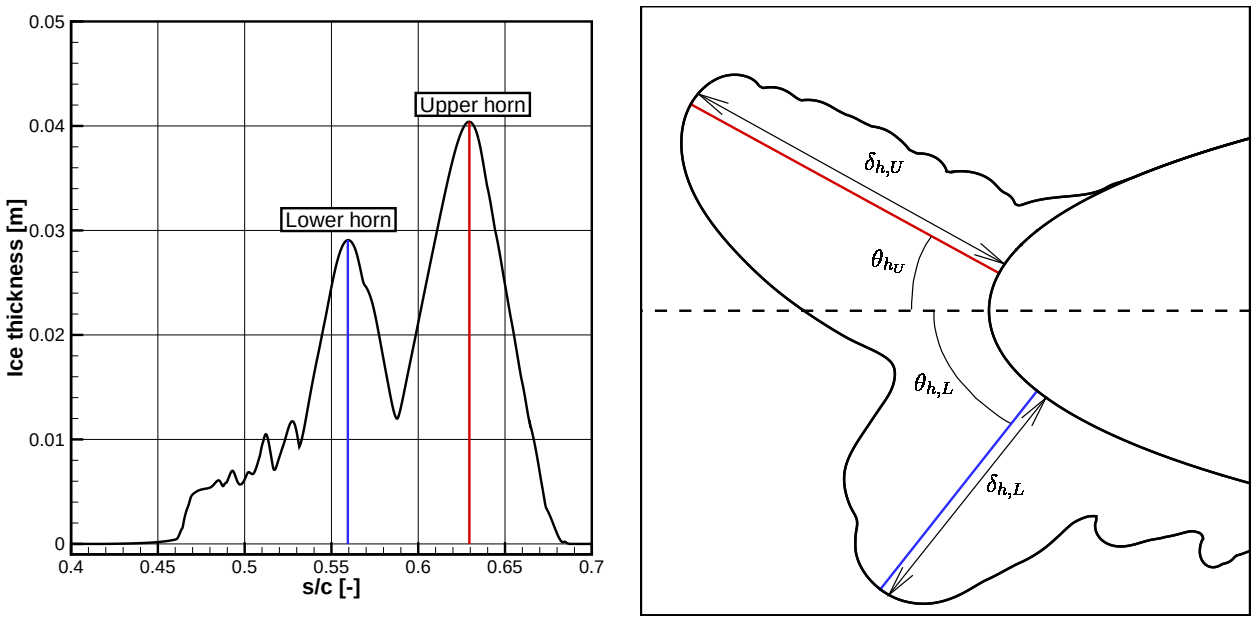
The comparative study is performed with five quantitative metrics :

- Ice horn length (δ_h);
- Ice horn angle (θ_h);
- Ice thickness distribution ($\delta_{ice} - s$);
- Ice coverage (Δs_{ice});
- Ice sectional area (Ω_{ice}).

Ice horns are generally related to glaze ice shapes where the horn is formed by the runback water mass flux which does not freeze in the stagnation region. Depending on the case parameters, the ice might accumulate into a short and thick horn or into a long and thin one, and its angle is closely related to its location on the airfoil as ice growth is normal to the surface by hypothesis. The freezing location on the airfoil depends mostly on the initial instants of the icing process, which explains why using multiple layers of ice affects the resulting ice shapes in glaze ice regime to the extent seen in §4.4.3. Therefore, the choice of numerical parameters, such as the number of ice layers, has a major impact on the results and parameter independent solutions should be sought whenever possible. It is less the case for rime ice regime, as the impingement water freezes on impact and the resulting horn is aligned with the streamwise direction. Nevertheless, ice horn length and angle are two metrics which can be used for both glaze and rime cases. The former might even feature two horns on the upper and lower sections of the airfoil, in which case these metrics are compared for each horn independently and the metrics are marked with U and L in subscripts to refer to the upper and lower horns respectively.

As computing ice horn length is akin to computing ice thickness, the latter is also included in this study. The ice horns and ice thickness metrics are indicators of how well the global ice accretion represents the multiple physical phenomena of the case, as it depends strongly on the convective heat transfer, runback water flow direction, droplet impingement and thermodynamic balances. For example, the location of minimal thickness between the upper and lower horns is representative of the stagnation point [109].

By making use of the B-splines that define both the ice shapes and the initial geometry, it is possible to compute the thickness distribution along the airfoil with a minimal distance method, as seen in Figure 6.1(a). The method relies on the analytical definition of the B-splines and uses inverse evaluations to find the parameter on the initial geometry B-splines



(a) Example of thickness distribution

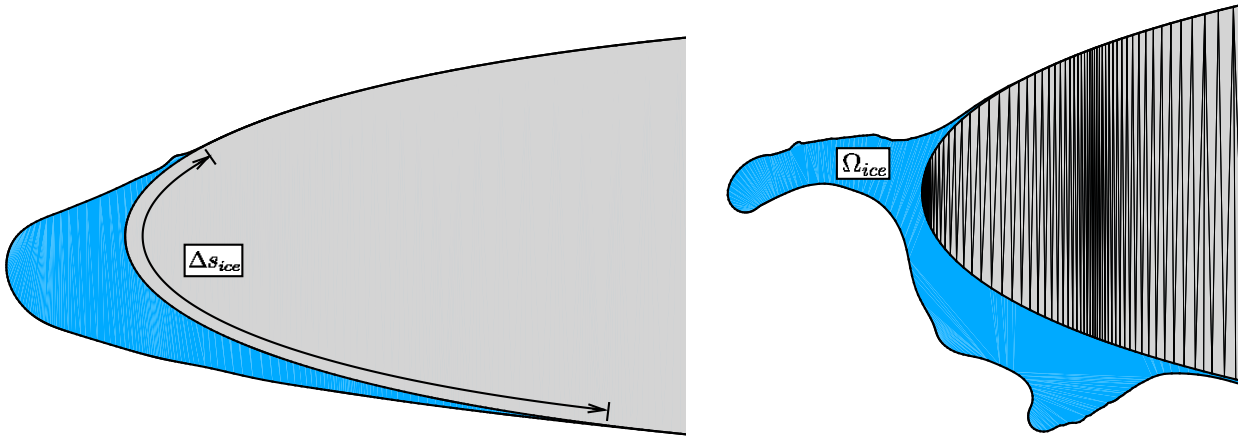
(b) Ice horn length and angle evaluation

Figure 6.1 Ice horn metrics and ice thickness distribution

which corresponds to the minimal distance from a tested point on the iced shape. This allows to remove the impact of the discretization error reported by Wright [109] which required the use of a special method for small ice thickness. The upper and lower horns lengths are then found by selecting the maximum ice thickness on both sides of the stagnation point and the horns angles are evaluated with respect to the horizontal axis, as seen on Figure 6.1(b). As already mentioned, the comparison of ice thickness distributions is more in the scope of a qualitative criterion, as it allows to compare easily if locations of maximum or minimum thickness in two ice shapes are close or not, whereas the ice horn length and angle are quantitative metrics which can be used to compute errors.

The ice thickness distribution is also used to find the ice coverage length by computing the curvilinear distance between the points where the ice section starts and ends, as seen in Figure 6.2(a). In rime ice conditions, the ice coverage should correspond to the water droplets impingement coverage, but this is not the case in glaze ice conditions where water flows on the surface and will freeze outside of the impingement region. Therefore, the ice coverage can be used to characterize the spreading of the water film on the surface, e.g. to know the section where anti-icing systems are needed.

The final quantitative metric is the ice sectional area. In this study, the exact internal area of the solids is computed by doing a constraint Delaunay triangulation. The ice sectional area is then obtained by making the difference between the iced airfoil internal area and the



(a) Ice coverage evaluation (b) Ice sectional area

Figure 6.2 Ice coverage and ice sectional area metrics

clean geometry internal area. This can be seen on Figure 6.2(b) where the ice sectional area is coloured in blue and the Delaunay triangulation of the clean airfoil is also shown. By comparing the global ice sectional area to an experimental ice sectional area, an error over the accumulated ice mass can be found.

The ice sectional area is useful during the geometry evolution simulations to act as a stopping criterion once the appropriate mass of ice is accumulated. Depending on the methods, a local ice area can also be used as a local mass conservation criterion. Furthermore, comparing layer by layer the actual ice area to the numerical prediction from the thermodynamic balances gives an estimate of the numerical mass conservation capability of the geometry evolution method.

6.3 Qualitative criteria

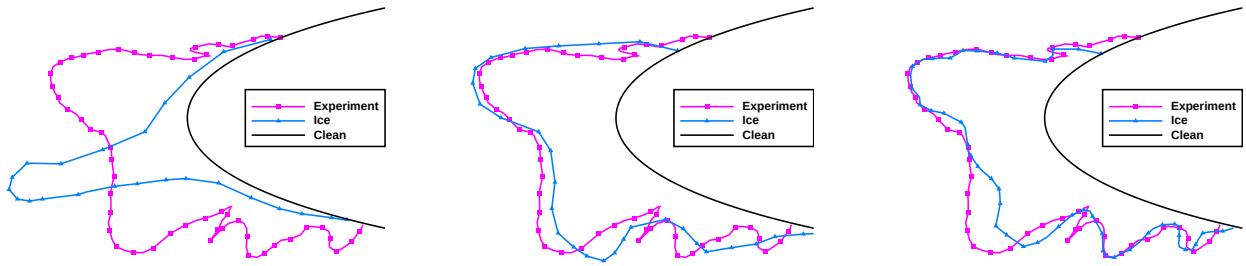
On top of the different quantitative metrics previously presented, the results shown later in this chapter are classified according to two qualitative criteria:

- The general shape;
- The presence of special features (small horns, surface roughness, etc.).

Since ice shapes are difficult to quantify with a specific metric, the literature introduces qualitative evaluations to assess if the results of an icing simulation are in agreement or not with the experimental data. Such evaluations compare different criteria similar to the

quantitative metrics mentioned earlier, but in a subjective manner to give a general opinion on the accordance between the results and the experiments. Three levels of classification can be defined for the general shape qualitative criterion: i) poor, ii) average and iii) good agreement with the experimental data. As examples, Figures 6.3(a) to 6.3(c) show fictive numerical ice shapes compared to an experimental ice shape. In Figure 6.3(a), the fictive shape is in poor agreement with the experiment as a thin horn is obtained whereas the experiment is wide and streamwise. On the contrary, Figure 6.3(b) shows a fictive ice shape which is in good agreement with the experiment, as the upper horn is very similar, the lower horn is only slightly misaligned and the ice coverage is also similar. An ice shape with average agreement would be in between, with only the upper horn matching the experiment, for example.

The second qualitative criterion is relevant in the case where special features such as small horns can be seen on the experimental data. In the case where there is any special feature in the experimental ice shape, then either this feature is captured by the numerical simulation or it is not. The experimental shape shown in Figures 6.3(a) to 6.3(c) contains smaller horns in the lower section of the ice and these can be classified as a special feature. The fictive ice shape in Figure 6.3(b) is in good agreement with the experiment, but it does not capture these smaller horns like the one shown in Figure 6.3(c).



(a) Poor agreement (b) Good agreement (c) Presence of smaller horns.

Figure 6.3 General ice shape qualitative evaluation

6.4 Experimental icing cases

In this section, the geometry evolution methods are compared against experimental data from Wright and Rutkowski [110]. A glaze ice (401) and a rime ice (405) cases are selected with the parameters presented in Table 6.1. The clean airfoil is a NACA0012 with a chord

of 0.5334m. Both cases are simulated in 1, 5, 10 and 20 layers with the algebraic method (§2.5.1), the tracked level-set method (Chapter 3) and the hyperbolic ice accretion method (Chapter 4).

Table 6.1 Icing cases parameters for the validation

Case		**Glaze Case 401**	**Rime Case 405**
AoA	[°]	3.5	3.5
Mach	[−]	0.315	0.324
Temperature	[K]	265.37	250.37
Pressure	[kPa]	100.0	100.0
LWC	[g/m^3]	0.55	0.55
MVD	[µm]	20	20
Icing Time	[s]	420	420
Roughness	[µm]	756.5	312.9

6.4.1 Glaze Ice: Case 401

The first experimental case is the glaze ice case 401 from Wright and Rutkowski [110] and the results for the three methods are compared to the experimental data in Figures 6.4(a) to 6.4(d). Table 6.2, Table 6.3 and Table 6.4 contain the quantitative metrics relative difference between the simulated shapes and the experimental shape for the algebraic, level-set and hyperbolic methods respectively. These tables also list the assessment of the qualitative criteria. The numerical area error is obtained by comparing the numerical ice area to the predicted ice area given by the thermodynamic solver at each layer, while the ice area error is obtained by comparing the numerical ice area to the experimental ice area.

The dimensional quantitative metrics obtained for all results, including the experimental data metrics, can be found in Appendix B. One important metric is the numerical area error which describes how a method globally conserves the mass. It can be seen that all of the methods have a low error below 0.5% in this aspect except for few results where the higher error is probably due to correction methods modifying the ice layers to ensure the geometries validity. As for the other quantitative metrics, no clear tendency can be found. In this case, the lower horn seems to be better captured in general than the upper horn, especially regarding the horn angle. However, the error obtained for the ice coverage remains similar for all results, which is expected as this metric is driven by the water film computed in the thermodynamic model.

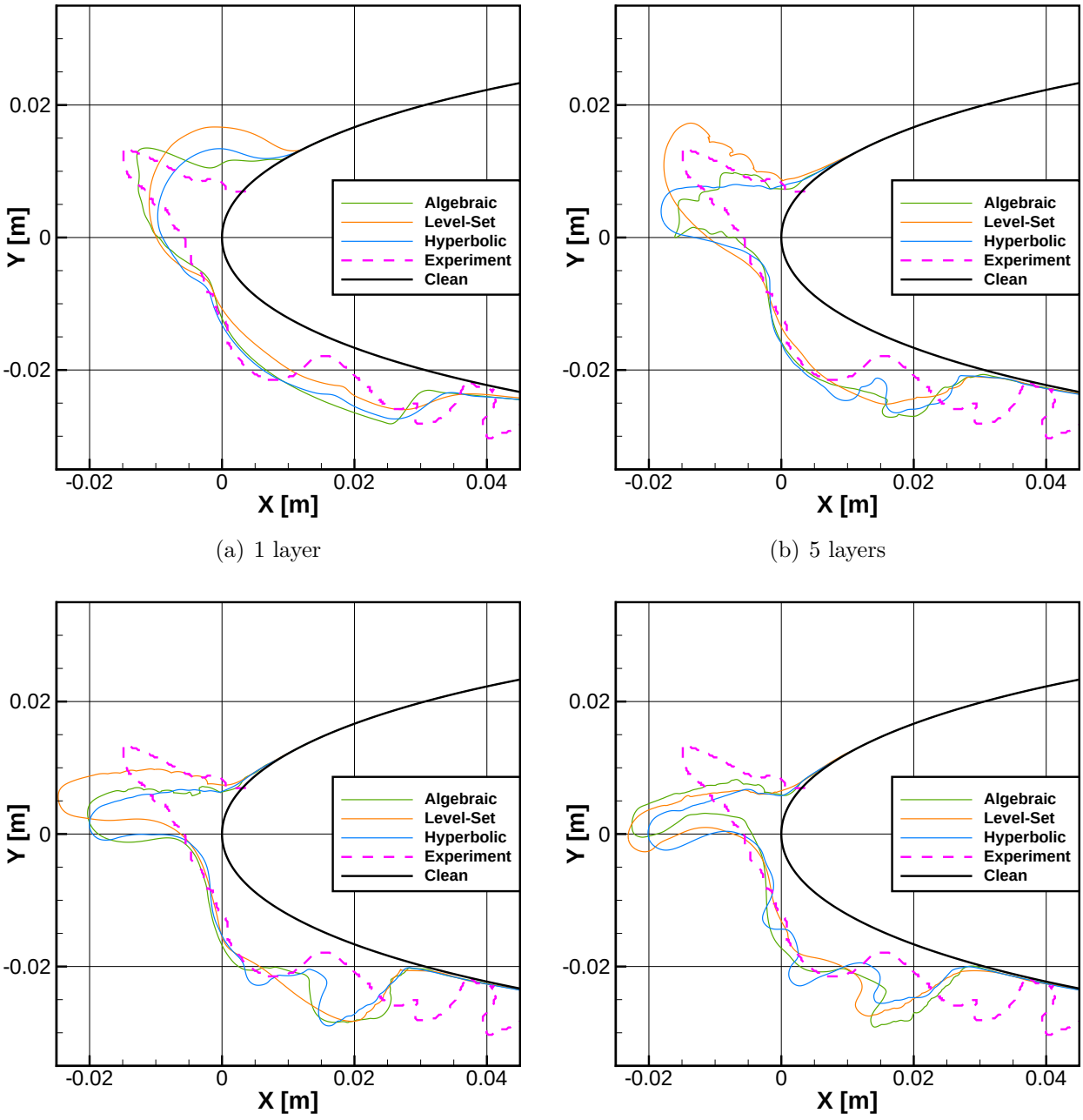
The main features of the experimental shape, illustrated with a pink dashed curve, are the well defined upper horn and the smaller horns located on the lower section of the airfoil.

Overall, the simulated ice shapes are converging toward a different solution than this experimental data, but some general characteristics are still similar to the experimental shape. For example, the region between the upper horn and the lower horn, as well as the lower horn itself and the limits of the upper horn, are well captured in general in all results. However, the upper horn bends toward a horizontal position as more layers are added and this causes the general agreement to the experiment to become poor. Only two results are somewhat in good agreement with this experimental shape: the single layer algebraic solution and the 5-layers level-set solution. However, these results are not independent from the number of layers and are only obtained by coincidence.

An interesting observation is that the level-set method seems to require more layers to converge toward a shape similar to what is obtained with the two other methods. Besides, it also required 20 layers for the level-set method to produce smaller horns in the lower region. Even then, all three methods are converging toward a similar shape with an almost horizontal upper horn and all three methods end up featuring the smaller horns in the lower region that can also be seen in the experimental data. The wrong orientation of the upper horn may be due to the convective heat transfer in the stagnation region being underestimated, which results in more water freezing in the horn regions. As the difference between the stagnation region and the horn regions is increased with successive layers, a highly concave zone is created around the stagnation point and the aerodynamic flow and droplets impingement are affected, thus amplifying the upper horn diverging phenomenon. A thorough study of the effect of convective heat transfer in the stagnation region is required to detail the observed discrepancies. Such study is beyond the scope of this thesis, but it is thought that the surface roughness might be in cause.

Table 6.2 Glaze case 401 quantitative metric errors : Algebraic method

Number of layers	1	5	10	20
Numerical area error	0.93 %	−3.92 %	−0.09 %	0.38 %
Ice area error	10.97 %	−7.68 %	0.06 %	−10.48 %
Upper horn length	−11.53 %	−12.77 %	11.11 %	22.53 %
Upper horn angle	4.66 %	−57.79 %	−76.79 %	−90.28 %
Lower horn length	−6.65 %	2.35 %	6.51 %	4.80 %
Lower horn angle	−18.87 %	1.84 %	7.03 %	6.00 %
Ice coverage	−29.32 %	−26.96 %	−27.45 %	−25.28 %
General shape	Good	Average	Poor	Poor
Smaller horns	No	Yes	Yes	Yes



(a) 1 layer

(b) 5 layers

(c) 10 layers

(d) 20 layers

Figure 6.4 Icing results for the NACA0012 glaze case 401

6.4.2 Rime Ice: Case 405

The second experimental case is the rime ice case 405 from Wright and Rutkowski [110] and Figures 6.5(a) to 6.5(d) show the comparison between the results obtained for the three methods and the experimental data. Table 6.5, Table 6.6 and Table 6.7 contain the quantitative metrics relative difference between the simulated shapes and the experimental shape

Table 6.3 Glaze case 401 quantitative metric errors : Tracked level-set method

Number of layers	1	5	10	20
Numerical area error	−0.02 %	0.50 %	−0.23 %	−2.32 %
Ice area error	9.92 %	12.94 %	6.57 %	−13.07 %
Upper horn length	−31.03 %	12.75 %	36.87 %	25.40 %
Upper horn angle	17.60 %	5.44 %	−62.96 %	−102.23 %
Lower horn length	−30.34 %	−1.49 %	14.82 %	27.28 %
Lower horn angle	−6.36 %	−6.58 %	−13.58 %	−9.50 %
Ice coverage	−27.30 %	−25.32 %	−27.01 %	−26.10 %
General shape	Average	Good	Average	Poor
Smaller horns	No	No	No	Yes

Table 6.4 Glaze case 401 quantitative metric errors : Hyperbolic method

Number of layers	1	5	10	20
Numerical area error	−1.99 %	0.08 %	0.17 %	0.14 %
Ice area error	7.76 %	−1.91 %	−9.34 %	−18.54 %
Upper horn length	−44.13 %	0.30 %	8.79 %	9.09 %
Upper horn angle	−22.29 %	−65.66 %	−88.33 %	−50.15 %
Lower horn length	−10.95 %	15.81 %	18.24 %	32.07 %
Lower horn angle	−2.48 %	−4.47 %	1.52 %	5.08 %
Ice coverage	−26.67 %	−25.78 %	−25.17 %	−23.00 %
General shape	Average	Average	Poor	Poor
Smaller horns	No	Yes	Yes	Yes

for the algebraic, level-set and hyperbolic methods respectively. These tables also list the assessment of the qualitative criteria. The dimensional quantitative metrics obtained for all results, including the experimental data metrics, can be found in Appendix B.

In this case, the numerical solutions obtained with all methods have a similar general shape for each number of layers. The only results which present differences for small details are the 5- and 10-layers solutions obtained with the algebraic method where small disturbances appear in the shape. This is caused by oscillations generated in the sub-layers of the algebraic method in the first ice layers which then affect the heat transfer and the runback water flow in the rest of the icing simulation, thus enhancing the oscillations. As a result of this, the 5-layers simulations have a very high error in terms of numerical ice area, as it can be seen in Table 6.5.

Overall, all methods are converging toward a solution that is very similar to the experimental result in terms of orientation, ice thickness and ice limits. Although this is the case quali-

tatively, this trend cannot be extended to the quantitative metrics, especially for the main horn angle. Note that for the main horn angle, the high error is in part due to the small experimental angle value. However, the error between the simulated ice area and the predicted area is low in almost all cases, which assess that the methods are globally conservative in terms of numerical mass up to 0.5% approximately. Furthermore, all methods are converging toward a very similar shape when enough layers are added, which means that they all have the same potential to represent the ice accretion phenomena in this case. This is expected for a rime case as i) the shape is prolonging the general airfoil shape in the streamwise direction and ii) there is no runback water on the surface to generate horns of ice.

Table 6.5 Rime case 405 quantitative metric errors : Algebraic method

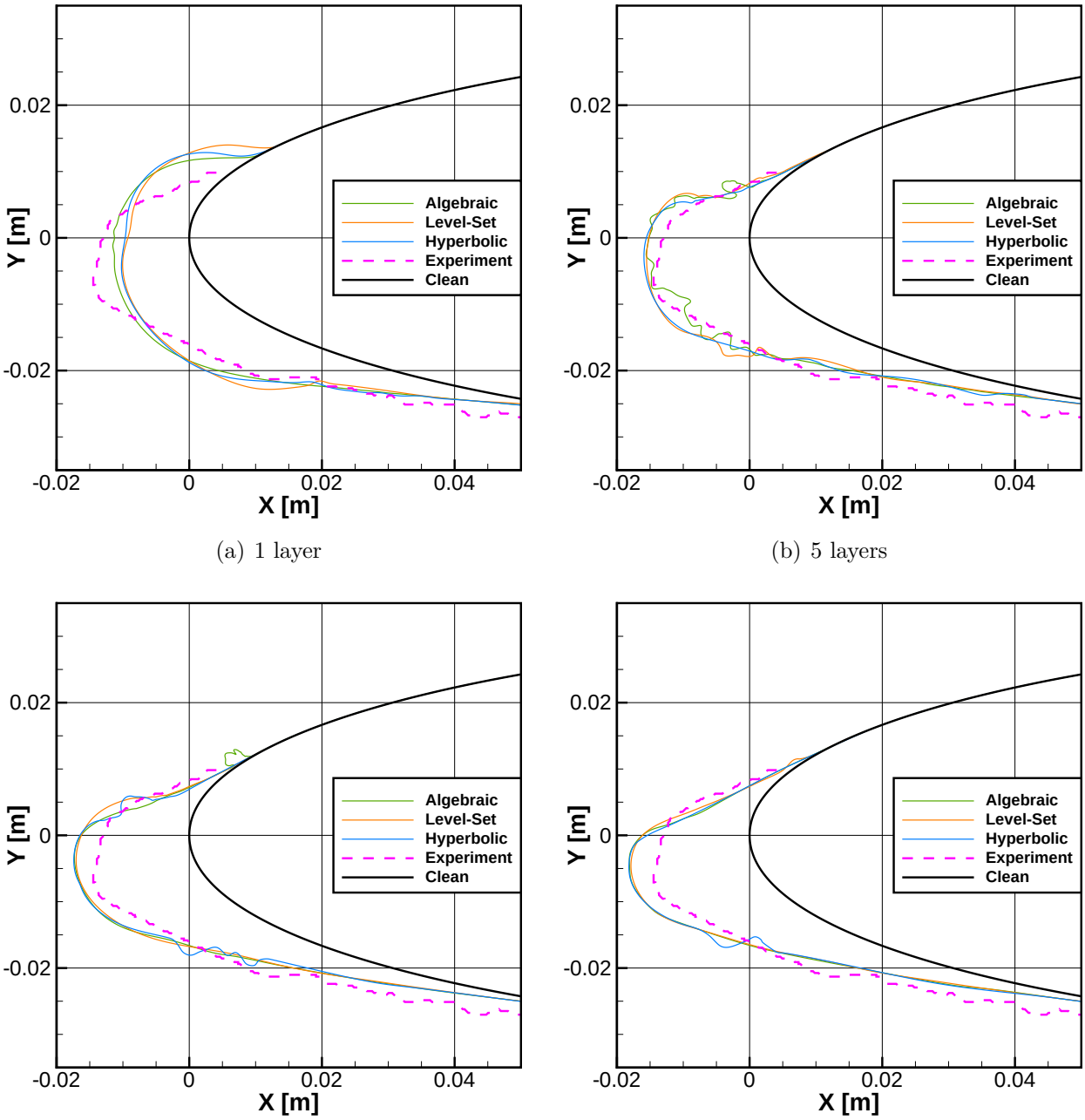
Number of layers	1	5	10	20
Numerical area error	1.04 %	−5.16 %	−0.01 %	−0.45 %
Ice area error	0.07 %	−10.55 %	−7.25 %	−8.67 %
Main horn length	−22.24 %	−6.67 %	12.74 %	19.83 %
Main horn angle	−64.43 %	12.09 %	51.18 %	27.86 %
Ice coverage	−11.57 %	8.62 %	8.53 %	9.54 %
General shape	Average	Average	Good	Good

Table 6.6 Rime case 405 quantitative metric errors : Tracked level-set method

Number of layers	1	5	10	20
Numerical area error	−0.01 %	−0.09 %	−0.25 %	−0.54 %
Ice area error	−0.97 %	−4.95 %	−6.61 %	−8.45 %
Main horn length	−28.40 %	3.19 %	11.74 %	17.95 %
Main horn angle	−81.91 %	2.92 %	36.22 %	30.98 %
Ice coverage	−8.37 %	9.43 %	8.13 %	10.82 %
General shape	Average	Good	Good	Good

Table 6.7 Rime case 405 quantitative metric errors : Hyperbolic method

Number of layers	1	5	10	20
Numerical area error	−0.15 %	−0.29 %	−0.20 %	−0.57 %
Ice area error	−1.11 %	−5.89 %	−7.86 %	−8.78 %
Main horn length	−27.82 %	4.29 %	13.91 %	20.15 %
Main horn angle	−94.86 %	23.48 %	40.60 %	30.23 %
Ice coverage	−8.44 %	5.50 %	2.13 %	4.65 %
General shape	Average	Good	Good	Good



(a) 1 layer

(b) 5 layers

(c) 10 layers

(d) 20 layers

Figure 6.5 Icing results for the NACA0012 rime case 405

6.5 Discussion

The previous section presented a partial validation of the three deterministic methods (algebraic, level-set and hyperbolic) studied in this chapter with respect to ice accretion. All three methods are converging toward similar solutions, when enough ice layers are used, for both the rime ice and the glaze ice cases. However, this converged shape does not correspond

to the experimental data in the case of the glaze ice case, as it can be seen on Figure 6.4(d). Also, the glaze case results indicate that the level-set method might require more ice layers than the two other methods to achieve convergence as the upper horn of the level-set shape changes significantly from the 10-layers simulations to the 20-layers simulation. Furthermore, the quantitative metrics in general do not show any tendency that one method would be better quantitatively than another. The only trend seen is that the methods are able to conserve the numerical ice mass up to approximately 0.5%, although the algebraic method is not always succeeding in this aspect, as seen in Table 6.5 for the 1- and 5-layers results. Therefore, it can be concluded that the three studied methods are approximately equivalent in terms of solving the ice geometry evolution physics when enough ice layers are simulated with each method. The main differences between the methods are thus in numerical implementation and robustness. Table 6.8 summarizes how the methods perform in different aspects related to these two categories.

In terms of computational cost, the algebraic method is the lightest method, as it requires only to solve algebraic equations on the surface in a few iterations (sub-layers). The hyperbolic method is marked as having a low numerical cost, as it requires to solve a 2-equations matrix system at each iteration which is more costly than the algebraic method. In terms of real time, no significant difference is seen between these two methods on 2D cases because both take less than one second to solve compared to hours for a full icing simulation, even for surface mesh made of around 1000 grid points. It can be expected that a more apparent difference would be seen for 3D cases, which explains the different ratings in Table 6.8. On the opposite, the level-set method requires very high computational resources due to the unsteady nature of the equation for which very small time steps are needed with very fine grids such as the one used to capture the boundary layer. It must also be mentioned again that the level-set equation is solved in the volume mesh, thus in a 2D topological grid instead of on a 1D one as for the other methods. Overall, a level-set ice accretion simulation can take up to 10 times the resources required for the aerodynamic flow simulation in a 1-layer simulation, but the ratio is lessened when more layers are used as the total physical time simulated with the level-set is divided among layers whereas the other modules are completely recomputed for each layers.

Regarding numerical ice mass conservation, the results presented in §6.4 show that all three methods can reach a sufficient degree of global conservation. This is due to the global mass active controls integrated in the level-set solver through the explicit tracking and in the hyperbolic solver by keeping track of the mesh area. No such active control is implemented into the algebraic method, but global mass conservation can be approximated by using enough sub-layers, although it does not ensure it as seen for some of the results in §6.4. However,

Table 6.8 Characteristics of the ice accretion methods

Method	Algebraic	Level-Set	Hyperbolic
Computational cost	Very low	Very high	Low
Mass conservation: global	Possible	Yes	Yes
Mass conservation: local	Possible	No	Yes
Robustness: foldings	Low	High	High
Robustness: front merge	Very low	High	Medium
Robustness: extraction	Medium	Low	High
3D extension	Difficult	Possible	Possible

it is possible to implement such an active control within an algebraic method, as stated in the literature [96]. As for local mass conservation, only the hyperbolic method in this study contains an active local control, whereas it would be possible to introduce it in a 2D algebraic method [35], but not in the level-set solver, which is why the definition of the icing velocity is important for the level-set method as discussed in §3.3.

Although the level-set method has some disadvantages, its major strength resides in the robustness with which it handles the front. As the front is defined implicitly, foldings cannot be produced and the method handles fronts merging without requiring any special treatments of the data. However, due to this implicit definition, extracting the final front to transform it into a B-spline or another geometry support is more difficult and requires to combine multiple methods in order to obtain a robust methodology. This aspect is less problematic in the algebraic and hyperbolic methods as the front is defined explicitly and can then be directly extracted, but both solvers require correction methods in the case where grid foldings or narrow concave regions appear in the solution. This is especially the case for the algebraic method where grid foldings always appear in concave regions, but it is less important for the hyperbolic ice accretion solver which makes use of smoothing mechanisms and of the Eikonal equation solution to handle both foldings and concave regions, as described in §4.2.1. As the algebraic method is less robust regarding grid foldings, the extracted solution can be invalid and the extraction robustness can only be marked as medium. Considering these different aspects, the hyperbolic ice accretion solver is the most robust method in this study.

A final aspect that combines both numerical implementation and robustness needs to be considered : the possibility to extend the method in 3D for multi-layer purposes. The algebraic method as presented in §2.5.1 could be extended in 3D, but it is very likely that robustness issues would further compromise the simulations and thus it would be difficult to use such method in a multi-layer 3D icing framework. Furthermore, even though the algebraic approach did not fail to compute the two cases presented in this study, it might have failed

in other cases due to grid folding and lack of general robustness. The level-set equation, being a PDE, is easily implemented in 3D, but would require a proper geometry extraction method to include it in a multi-layer icing framework. However, the computational cost of the method as implemented in NSCODE-ICE is high and a more lightweight implementation would be required for a realistic 3D extension. As this equation is the subject of many studies in the literature, different solving approaches can be easily designed for the icing application and thus 3D extension is still possible. The hyperbolic method is robust and requires low computational resources, which are two advantages for a 3D extension. Furthermore, the method is based on mesh generation equations which are used to generate 3D grids [25], thus it is implicit that the equations are also working well in 3D for ice accretion purpose. The only downside of the equations is that they are developed for structured grids only, with a finite difference method. Therefore, an unstructured 3D extension would require more developments.

6.6 Conclusion

In this chapter, the three deterministic approaches presented in this study, the algebraic, level-set and hyperbolic methods, are compared on experimental ice cases. Such a comparison does not exist in the literature to the best of the author knowledge, with different geometry evolution methods being used within the same ice accretion solver. The quantitative metrics and qualitative criteria used in this comparative study are detailed with some justifications regarding their selection. Then, the results obtained on two experimental cases are presented.

By adding enough layers, the three methods are converging toward a similar shape in both cases, but for the glaze ice case it does not correspond to the experimental shape. The quantitative metrics do not allow to identify trends which would highlight differences between the methods, except for the numerical ice area error which show that all three methods can conserve the global ice mass which is predicted by the thermodynamic model.

A discussion then follows to compare the methods in aspects related to numerical implementation and robustness in view of a possible extension in three dimensions. The algebraic method has a very low robustness and can be discarded in this scope. The level-set method is robust and can be actively controlled to conserve the global ice mass, but it requires high computational resources in the present implementation. The hyperbolic method is identified as the most robust method, but also as the one allowing the best ice mass conservation by combining global and local controls. Furthermore, it does not require much computational resources and the literature shows that the 3D extension is possible for structured grids.

CHAPTER 7 CONCLUSION

This final chapter presents the conclusions drawn in the present study with a focus on the elements which are thought to advance the current knowledge. Limitations of the presented methods and of the study itself are then mentioned, followed by recommendations for future work.

The main objective that drives this entire research is to model complex ice accretion shapes accurately in a fully automated multi-layer icing software with an appropriate set of methods for the geometry evolution process and the transition phase between each ice layers. The developments are done in 2D, but the purpose is to eventually extend these methods in 3D, which requires robustness in priority and efficiency as a secondary consideration. The different modules interacting in the complete numerical icing process are first detailed to provide a background to this research and the geometry is identified as being the critical part of the software to improve in order to develop a fully automated multi-layer process. Thus, a continuous geometry representation in the form of B-splines is added to the framework to ensure robust transitions from an evolved ice front solution to a new ice geometry which can easily be used to generate new grids for the next ice layer.

Two deterministic methods are then developed to evolve the ice front in an attempt to handle robustly the front topological variations, such as merging fronts, while also conserving the ice mass. The level-set ice interface capturing method is improved from the literature with an explicit tracking which allows to globally conserve the mass. In addition, a mathematical formulation for the velocity driving the front displacement is also studied to formalize the model for mass conservation. In a different approach, a hyperbolic ice accretion method is developed based on the hyperbolic mesh generation equations by modifying the source term in an original manner to include a knowledge on the global geometry via the use of the Eikonal equation solution. This hyperbolic method contains global and local controls on mass conservation and benefits from robust smoothing mechanisms developed for the grid generation in the literature.

A non-deterministic approach is then adapted from the literature to perform multi-layer icing. It builds on the already existing Eulerian modules from the software to compute the different probabilities used to introduce stochasticity in the icing process. Finally, a comparative study is performed between the two deterministic front evolution methods developed in this study and the algebraic geometry evolution method generally used in the literature. The methods are compared quantitatively and qualitatively on two experimental icing cases and the results

obtained for the 20-layers simulations are all converging toward a similar shape.

In the end, numerical implementation and robustness considerations are studied to determine that the hyperbolic ice accretion method is, to date, the most suited method for an extension in three dimensions. Moreover, defining the geometry as the critical component of the icing simulation and adding B-splines allowed to achieve the objective regarding automated multi-layer simulations.

7.1 Advancement of knowledge

In this research, different methods to evolve the ice accreted geometry are studied and are based in part on the literature. In the different chapters, details are given on original modifications that aim at robustness of the process and ice mass conservation. This section highlights these contributions with justification on their importance.

7.1.1 B-splines geometry

The integration of B-splines as a geometry support to handle the interface between the geometry evolution method and the mesh generation process is an important original aspect of this thesis as it increases the robustness and it allows a better automation of the multi-layer icing process, as detailed in Chapter 2. Although such B-splines and other continuous geometry supports have already been seen in icing literature [97, 78], an original approximation method is introduced [16] to generate the B-splines with a local error driven algorithm which allows to better capture local geometry features. The integration of B-splines in the framework has multiple effects in the simulations: i) a more robust mesh generation, ii) more accurate surface discretization strategies and iii) seamless post-processing capacities.

7.1.2 Level-set method

Chapter 3 presented original developments on the level-set method applied for ice accretion with a focus on front data extraction and global mass conservation. A single method, namely the explicit tracking method, allows to perform these two tasks with a sufficient level of robustness in general. Furthermore, to ensure ice mass conservation in the level-set equation itself, a modification to the formulation for the interface velocity from the literature [70] is suggested and studied. In addition, the work extends single layer level-set results [70] to multi-layer capability. The level-set method handles automatically fronts merging, which is a problematic phenomenon seen in ice accretion, although it is not shown in the experimental cases studied in this work. Therefore, this study contributes to the increase in technological

readiness level of the level-set approach for ice accretion by providing a methodology to ensure ice mass conservation and to perform multi-layer simulations with an implicit interface capturing method.

7.1.3 Hyperbolic method

An original geometry evolution method for ice accretion is presented in Chapter 4 and is based on the hyperbolic mesh generation equations [25], resulting in a mesh-like geometry evolution. The source term of the equations is modified in two ways: i) by directly using the ice volume accretion mapping from the thermodynamic model to drive the ice sub-layers expansion and ii) by adding a new source term linked to the Eikonal equation gradient. The second modification allows the sub-layers to deviate from the orthogonal constraint in the case where the sub-layers are generated toward the initial geometry, which would introduce a surface grid folding. Furthermore, as the method generates a grid in order to displace the ice front, local and global mass conservation controls are easily added. The results obtained with the hyperbolic ice accretion method are promising and the robustness of the method suggests that it represents a potential choice for 3D extension.

7.1.4 Stochastic behaviour

The above improvements in icing front evolution methods could enable multi-layer capability in the single step non-deterministic method of Szilder and Lozowski [95]. To demonstrate such capability, the current work reinterprets the work of Szilder and Lozowski [95] by taking into account the strengths of the NSCODE-ICE framework, e.g. the Eulerian droplet solver and the overset grids capability. First, a novel advancing front formulation allows to drastically improve the efficiency of the method by building Cartesian cells only where needed around the geometry. Second, using the geometry modelling approach described in Chapter 2, multi-layer simulations are performed with the stochastic ice accretion method, which represents an original contribution to the field. It is shown that even though the method is stochastic, the curve averaged from multiple runs converges toward a solution when enough ice layers are used.

7.1.5 Comparative study

Finally, Chapter 6 presented an original comparative study where different deterministic geometry evolution methods are used within the same numerical framework for further validation. The quantitative metrics used to compare the numerical shapes to the experimental

data do not allow to conclude on the validity of the results, but the comparison of the iced area with the predicted ice area shows that all the studied methods conserve the mass globally up to 0.5% in most cases. The methods are compared regarding their strength and flaws with respect to numerical implementation and robustness based on observation made during the study. Although heuristic, this comparison could be used to make a choice regarding the method to implement in a new ice accretion software.

7.2 Limitations

The different methods presented in this study all have different limitations, with the most critical listed below.

- The geometry approximation method sometimes results in a warped B-spline that, although valid in itself, produces a geometry with narrow concave regions on which even the hyperbolic mesh generation method produces cells with negative volumes.
- The level-set method is expensive in computational resources and the front coordinates extraction might fail in convex regions in the case where the discretization is too coarse as the gradient that drives the explicit tracking is wrongly computed near the surface.
- The re-initialization equation in the level-set module is a simple one which conserves the volume by freezing some cells around the interface during the re-initialization process in a heurisitc manner, whereas more advanced algorithms can be found in the literature to re-initialize the solution while still conserving the mass.
- Even though the source term of the hyperbolic ice accretion approach allows to avoid grid foldings in most cases, it might still fail if the concavity is too high, which would then require a method to merge the fronts.
- The stochastic approach is limited to rime ice cases for now since the runback water process does not include a displacement memory for the water elements, resulting in a wrong spreading of the water and of the ice.
- Interpolations techniques were used in various methods, such as in the stochastic ice accretion, but were not studied to evaluate their diffusive effects on the ice accretion.
- The partial validation made in the comparative study includes only two cases, which is not representative of a comprehensing validation.

On a global scale, the study is limited to 2D models and results even though the purpose is to develop a method which could be extended to a 3D solver. Furthermore, the effect of

numerical parameters, such as the surface roughness, is not studied even though it is known that it has a major impact on the numerical icing simulation results. For example, the effects of applying roughness due to ice on the complete airfoil instead of only on the iced surface, as mentioned in §2.2.3, are not studied and might explain in part the discrepancies between the numerical ice shapes and the experimental ones.

7.3 Recommendations

The limitations discussed previously need to be addressed through further research and some solutions are proposed in this section.

- The B-spline approximation method could be improved by forcing a tangential continuity with the underlying curve in which the ice front B-spline is merged. This is already done in the standard fitting method of EGADS and could be exported into the approximation.
- Reducing the computational cost of the level-set approach could be done by using a direct unsteady approach instead of a dual time-stepping approach.
- To solve the level-set method front extraction issue encountered in this study, a more robust surface discretization method would have to ensure the presence of fine surface elements in convex regions.
- A method to merge fronts automatically should be added to the hyperbolic method to improve the robustness, even though it was not found to be necessary in this study.
- The stochastic approach is still implemented with a heuristic thinking and would require a more thorough examination of how droplets dispersed randomly in a cloud impacting the surface.

Globally, the method with the highest potential for multi-layer 3D extension is probably the hyperbolic ice accretion approach and efforts should be invested to validate it and study how the parameters affect the results.

REFERENCES

[1] ABDOLLAHI, V., HABASHI, W. G., and FOSSATI, M., "Hybrid quasi molecular-continuum modeling of supercooled large droplet dynamics for in-flight icing conditions", in *54^{th} AIAA Aerospace Sciences Meeting*. AIAA Paper 2016-0061, Jan. 2016. DOI: 10.2514/6.2016-0061

[2] ALIAGA, C. N., AUBÉ, M. S., BARUZZI, G. S., and HABASHI, W. G., "FENSAP-ICE-Unsteady: Unified in-flight icing simulation methodology for aircraft, rotorcraft, and jet engines", *Journal of Aircraft*, vol. 48, no. 1, pp. 119–126, 2011. DOI: 10.2514/1.C000327

[3] ATTARZADEH, R., YEGANEHDOUST, F., and DOLATABADI, A., "Icephobic performance of superhydrophobic coatings: A numerical analysis", in *7^{th} European Conference for Aeronautics and Space Sciences*. EUCASS Paper 2017-668, Jul.. 2017. DOI: 10.13009/EUCASS2017-668

[4] AUSAS, R. F., DARI, E. A., and BUSCAGLIA, G. C., "A geometric mass-preserving redistancing scheme for the level set function", *International Journal for Numerical Methods in Fluids*, vol. 65, no. 8, pp. 989–1010, 2011. DOI: 10.1002/fld.2227

[5] BANSMER, S. E. and BAUMERT, A., "From high altitude clouds to an icing wind tunnel: en route to understand ice crystal icing", in *7^{th} European Conference for Aeronautics and Space Sciences*. EUCASS Paper 2017-665, Jul.. 2017. DOI: 10.13009/EUCASS2017-665

[6] BARTH, T. and JESPERSEN, D., "The design and application of upwind schemes on unstructured meshes", in *27^{th} Aerospace Sciences Meeting*. AIAA Paper 1989-366, Jan. 1989. DOI: 10.2514/6.1989-366

[7] BEAUGENDRE, H., "A PDE-based 3D approach to in-flight ice accretion", Ph. D. Thesis, McGill University, Montreal, Jun. 2003.

[8] BLAZEK, J., *Computational Fluid Dynamics: Principles and Application*, 3rd ed. Elsevier, 2015.

[9] BOURGAULT, Y., HABASHI, W. G., DOMPIERRE, J., and BARUZZI, G. S., "A finite element method study of eulerian droplets impingement models", *International Journal for Numerical Methods in Fluids*, vol. 29, pp. 429–449, 1999. DOI: 10.1002/(SICI)1097-0363(19990228)29:4<429::AID-FLD795>3.0.CO;2-F

[10] BOURGAULT, Y., BEAUGENDRE, H., and HABASHI, W. G., "Development of a shallow-water icing model in FENSAP-ICE", *Journal of Aircraft*, vol. 37, no. 4, pp. 640–646, 2000. DOI: 10.2514/6.1999-246

[11] BOURGAULT-CÔTÉ, S. and LAURENDEAU, E., "Two-dimensional/infinite swept wing ice accretion model", in *53rd AIAA Aerospace Sciences Meeting*. AIAA Paper 2015-535, Jan. 2015. DOI: 10.2514/6.2015-0535

[12] BOURGAULT-CÔTÉ, S. and LAURENDEAU, E., "Development of a morphogenetic ice accretion solver using overset grids", in *CFD Society of Canada 24th Annual Conference*. CFDSC, Jun. 2016.

[13] BOURGAULT-CÔTÉ, S. and LAURENDEAU, E., "Ice accretion geometry evolution on airfoils with level-set method", in *63rd CASI Aeronautics Conference*. CASI, May 2017.

[14] BOURGAULT-CÔTÉ, S., HASANZADEH, K., LAVOIE, P., and LAURENDEAU, E., "Multi-layer icing methodologies for conservative ice growth", in *7th European Conference for Aeronautics and Space Sciences*. EUCASS Paper 2017-258, Jul.. 2017. DOI: 10.13009/EUCASS2017-258

[15] BOURGAULT-CÔTÉ, S., DOCAMPO-SÀNCHEZ, J., and LAURENDEAU, E., "Multi-layer ice accretion simulations using a level-set method with B-Spline representation", in *56th AIAA Aerospace Sciences Meeting*. AIAA Paper 2018-1835, Jan. 2018. DOI: 10.2514/6.2018-1835

[16] BOURGAULT-CÔTÉ, S., DOCAMPO-SÀNCHEZ, J., and LAURENDEAU, E., "Multi-layer airfoil ice accretion simulations using a level-set method with B-Spline representation", *AIAA Journal*, submitted 2018, accepted with revisions on January 5, 2019. Manuscript ID 2018-09-J057905.

[17] BOURGAULT-CÔTÉ, S., "Simulation du givrage sur ailes en flèche par méthodes RANS/Eulérienne quasi stationnaires", Master Thesis, Polytechnique Montréal, Montréal, May 2015.

[18] BRAGG, M. B., BROEREN, A. P., and BLUMENTHAL, L. A., "Iced-airfoil aerodynamics", *Progress in Aerospace Sciences*, vol. 41, no. 5, pp. 323–362, 2005. DOI: 10.1016/j.paerosci.2005.07.001

[19] BRAHIMI, M. T., TRAN, P., and PARASCHIVOIU, I., "Numerical simulation and thermodynamic analysis of ice accretion on aircraft wings", École Polytechnique de Montréal, Tech. Rep., May 1994.

[20] BRAVIN, M., STRAPP, J. W., and MASON, J. G., "An investigation into location and convective lifecycle trends in an ice crystal icing engine event database", in *SAE 2015 Internation Conference on Icing of Aircraft, Engines and Structures*. SAE Technical Paper 2015-01-2130, Jun. 2015. DOI: 10.4271/2015-01-2130

[21] BUREAU D'ÊNQUETE ET D'ANALYSE POUR LA SÉCURITÉ DE L'AVIATION CIVILE (BEA), "Final Report on the accident on 1st June 2009 to the Airbus A330-203 registered F-GZCP operated by Air France flight AF 447 Rio de Janeiro - Paris", Ministère de l'Écologie, du Développement durable, des Transports et du Logement, Tech. Rep., Jul.. 2012.

[22] BUTNARASU, M., HABASHI, W. G., and FOSSATI, M., "Optimization of the morphogenetic approach for in-flight icing", in *53rd AIAA Aerospace Sciences Meeting*. Jan.: AIAA Paper 2015-34, 2015. DOI: 10.2514/6.2015-0034

[23] CAPGEMINI GROUP. OpenCASCADE. https://www.opencascade.com.

[24] CHAN, W. M., "Hyperbolic methods for surface and field grid generation", in *Handbook of Grid Generation*, 1st ed., THOMPSON, J. F., SONI, B. K., and WEATHERILL, N. P., Eds. CRC Press, 1999.

[25] CHAN, W. M. and STEGER, J. L., "Enhancements of a three-dimensional hyperbolic grid generation scheme", *Applied Mathematics and Computation*, vol. 51, no. 2, pp. 181–205, 1992. DOI: 10.1016/0096-3003(92)90073-A

[26] CHAWNER, J. R., MICHAL, T. R., SLOTNICK, J. P., and RUMSEY, C. L., "Summary of the 1st AIAA geometry and mesh generation workshop (GMGW-1) and future

plans", in *56th AIAA Aerospace Sciences Meeting*, series AIAA SciTech Forum. AIAA Paper 2018-0128, Jan. 2018. DOI: 10.2514/6.2018-0128

[27] DE BOOR, C., *A Practical Guide to Splines*, series Applied Mathematical Sciences. Springer New York, 2001.

[28] DIEBOLD, J. M., BROEREN, A. P., and BRAGG, M., "Aerodynamic classification of swept-wing ice accretion", in *Fluid Dynamics and Co-located Conferences*. AIAA Paper 2013-2825, Jun. 2013. DOI: 10.2514/6.2013-2825

[29] DURST, F., MILOJEVIC, D., and SCHÖNUNG, B., "Eulerian and lagrangian predictions of particulate two-phase flows: a numerical study", *Applied Mathematics Modelling*, vol. 8, pp. 101–115, 1984. DOI: 10.1016/0307-904X(84)90062-3

[30] FEDERAL AVIATION ADMINISTRATION, *Aviation Weather for pilots and flight operations personnel : Chapter 10 - Icing*. FAA Flight Standards Service and the National Weather Service (NWS), 1975. Retrieved from URL: https://www.faa.gov/documentLibrary/media/Advisory_Circular/AC%2000-6A%20Chap%2010-12.pdf

[31] FEDERAL AVIATION ADMINISTRATION. Eletronic Code of Federal Regulations, Appendix C to Part 25. Last accessed: 2018-08-16. Retrieved from URL: https://www.ecfr.gov/cgi-bin/text-idx?SID=bbf24e9c782b2057583037ed7c2efe26&mc=true&node=pt14.1.25&rgn=div5#ap14.1.25.0000_0nbspnbspnbsp.c

[32] FEDERAL AVIATION ADMINISTRATION. Eletronic Code of Federal Regulations, Appendix O to Part 25. Last accessed: 2018-08-16. Retrieved from URL: https://www.ecfr.gov/cgi-bin/text-idx?SID=bbf24e9c782b2057583037ed7c2efe26&mc=true&node=pt14.1.25&rgn=div5#ap14.1.25.0000_0nbspnbspnbsp.o

[33] FEDERAL AVIATION ADMINISTRATION. Eletronic Code of Federal Regulations, Appendix D to Part 33. Last accessed: 2018-08-16. Retrieved from URL: https://www.ecfr.gov/cgi-bin/text-idx?SID=bbf24e9c782b2057583037ed7c2efe26&mc=true&node=pt14.1.33#ap14.1.33.0000_0nbspnbspnbsp.d

[34] FORTIN, G., ILINCA, A., LAFORTE, J.-L., and BRANDI, V., "New roughness computation method and geometric accretion model for airfoil icing", *Journal of Aircraft*, vol. 41, no. 1, pp. 119–127, 2004. DOI: 10.2514/1.173

[35] FORTIN, G., LAFORTE, J.-L., and ILINCA, A., "Heat and mass transfer during ice accretion on aircraft wings with an improved roughness model", *International Journal of Thermal Sciences*, vol. 45, pp. 595–606, 2006. DOI: 10.1016/j.ijthermalsci.2005.07.006

[36] FOULADI, H., ALIAGA, C. N., and HABASHI, W. G., "Quasi-unsteady icing simulation of an oscillating airfoil", in *7th AIAA Atmospheric and Space Environments Conference*. AIAA Paper 2015-3020, Jun. 2015. DOI: 10.2514/6.2015-3020

[37] FROLKOVIČ, P., MIKULA, K., and URBÁN, J., "Semi-implicit finite volume level set method for advective motion of interfaces in normal direction", *Applied Numerical Mathematics*, vol. 95, pp. 214–228, sep 2015. DOI: 10.1016/j.apnum.2014.05.011

[38] FUJIWARA, G. E. C., WOODARD, B. S., WIBERG, B. D., MORTONSON, A. J., and BRAGG, M. B., "A hybrid airfoil design method for icing wind tunnel tests", in *5th AIAA Atmospheric and Space Environments Conference*. AIAA Paper 2013-2826, Jun. 2013. DOI: 10.2514/6.2013-2826

[39] GHASEMI, S., MOSAHEBI, A., and LAURENDEAU, E., "A two-dimensional/infinite swept wing navier-stokes solver", in *52nd AIAA Aerospace Sciences Meeting*. AIAA Paper 2014-557, Jan. 2014. DOI: 10.2514/6.2014-0557

[40] HAIMES, R. and DANNENHOFFER, J., "The Engineering Sketch Pad: A solid-modeling, feature-based, web-enabled system for building parametric geometry", in *21st AIAA Computational Fluid Dynamics Conference*. AIAA Paper 2013-3073, Jun. 2013. DOI: 10.2514/6.2013-3073

[41] HARASE, S., "On the $\mathbb{F}_2$-linear relations of Mersenne Twister pseudorandom number generators", *Mathematics and Computers in Simulation*, vol. 100, pp. 103–113, 2014. DOI: 10.1016/j.matcom.2014.02.002

[42] HASANZADEH, K., LAURENDEAU, E., and PARASCHIVOIU, I., "Quasi-steady convergence of multi-steps navier-stokes icing simulations", *Journal of Aircraft*, vol. 50, no. 4, pp. 1261–1274, 2013. DOI: 10.2514/1.C032197

[43] HASANZADEH, K., MOSAHEBI, A., LAURENDEAU, E., and PARASCHIVOIU, I., "Validation and verification of multi-steps icing calculation using CANICE2D-NS

code", in *31st AIAA Applied Aerodynamics Conference*. AIAA Paper 2013-2671, Jun. 2013. DOI: 10.2514/6.2013-2671

[44] HASANZADEH, K., LAURENDEAU, E., and PARASCHIVOIU, I., "Adaptive curvature control grid generation algorithms for complex glaze ice shapes rans simulations", in *53rd AIAA Aerospace Sciences Meeting*. AIAA Paper 2015-914, Jan. 2015. DOI: 10.2514/6.2015-0914

[45] HASANZADEH, K., LAURENDEAU, E., and PARASCHIVOIU, I., "Grid-generation algorithms for complex glaze-ice shapes reynolds-averaged navier-stokes simulations", *AIAA Journal*, vol. 54, no. 3, pp. 847–860, 2016. DOI: 10.2514/1J054076

[46] HASANZADEH LASHKAJANI, K., "Reynolds-averaged navier-stokes based ice accretion for aircraft wings", Ph. D. Thesis, Polytechnique Montreal, Montreal, Dec. 2015.

[47] HONSEK, R., HABASHI, W. G., and AUBÉ, M. S., "Eulerian modeling of in-flight icing due to supercooled large droplets", *Journal of Aircraft*, vol. 45, no. 4, pp. 1290–1296, Jul.. 2008. DOI: 10.2514/1.34541

[48] ISHIKAWA, H., TANAKA, K., MAKINO, Y., and YAMAMOTO, K., "Sonic-boom prediction using Euler CFD codes with structured/unstructured overset method", in *27th Congress of International Council of the Aeronautical Sciences*. ICAS, Sep. 2010.

[49] JAMESON, A., SCHMIDT, W., and TURKEL, E., "Numerical solutions of the euler equations by finite volume methods using runge-kutta time-stepping schemes", in *14th Fluid and Plasma Dynamics Conference, Fluid Dynamics and Co-located Conferences*. AIAA Paper 1981-1259, Jun. 1981. DOI: 10.2514/6.1981-1259

[50] KIM, I., BACHCHAN, N., and PEROOMIAN, O., "Supercooled large droplet modeling for aircraft icing using an eulerian-eulerian approach", *Journal of Aircraft*, vol. 53, no. 2, pp. 487–500, 2016. DOI: 10.2514/1.C033506

[51] KINSEY, D. W. and BARTH, T. J., "Description of a hyperbolic grid generating procedure for arbitrary two-dimensional bodies", AIR FORCE WRIGHT AERONAUTICAL LABS, Tech. Rep., 1994.

[52] LAVOIE, P., LAURENDEAU, E., PENA, D., and HOARAU, Y., "Comparison of thermodynamic models for ice accretion on airfoils", *International Journal of Numerical Methods for Heat and Fluid Flow*, 2017. DOI: 10.1108/HFF-08-2016-0297

[53] LAVOIE, P., BOURGAULT-CÔTÉ, S., and LAURENDEAU, E., "Numerical algorithms for infinite swept wing ice accretion", *Computers & Fluids*, vol. 161, pp. 189–198, Jan. 2018. DOI: 10.1016/j.compfluid.2017.11.015

[54] LAVOIE, P., "A numerical model simulating thin water films on swept wings in icing condition", Master Thesis, Polytechnique Montréal, Apr.. 2017.

[55] LEARY, W. M., *We Freeze to Please: A History of NASA's Icing Research Tunnel and the Quest for Flight Safety*, series The NASA history series. NASA, 2002.

[56] LÉVESQUE, A. T., PIGEON, A., DELOZE, T., and LAURENDEAU, E., "An overset grid 2D/infinite swept wing URANS solver using recursive cartesian bucket method", in *53rd AIAA Aerospace Sciences Meeting*. AIAA Paper 2015-912, Jan. 2015. DOI: 10.2514/6.2015-0912

[57] MCCLAIN, S. T., VARGAS, M., TSAO, J.-C., BROEREN, A. P., and LEE, S., "Ice accretion roughness measurements and modeling", in *7th European Conference for Aeronautics and Space Sciences*. EUCASS Paper 2017-555, Jul.. 2017. DOI: 10.13009/EUCASS2017-555

[58] MENTER, F. R., "Two-equation eddy-viscosity turbulence models for engineering applications", *AIAA Journal*, vol. 32, no. 8, pp. 1598–1605, 1994. DOI: 10.2514/3.12149

[59] MESSINGER, B., "Equilibrium temperature of an unheated icing surface as a function of airspeed", *Journal of the Aeronautical Sciences*, vol. 1, no. 20, pp. 29–42, Jan. 1953. DOI: 10.2514/8.2520

[60] MONTREUIL, E., CHAZOTTES, A., GUFFOND, D. *et al.*, "ECLIPPS: 1. three-dimensional CFD prediction of the ice accretion", in *1st AIAA Atmospheric and Space Environments Conference*. AIAA Paper 2009-3969, Jun. 2009. DOI: 10.2514/6.2009-3969

[61] MOSAHEBI, A. and LAURENDEAU, E., "Convergence characteristics of fully and loosely coupled numerical approaches for transition models", *AIAA Journal*, vol. 53, no. 5, pp. 1399–1404, 2015. DOI: 10.2514/1.J053722

[62] MYERS, T. G., "Extension to the messinger model for aircraft icing", *AIAA Journal*, vol. 39, no. 2, pp. 211–218, 2001. DOI: 10.2514/2.1312

[63] MYERS, T. G., CHARPIN, J. P. F., and THOMPSON, C. P., "Slowly accreting ice due to supercooled water impacting on a cold surface", *Physics of Fluids*, vol. 14, no. 1, pp. 240–257, 2002. DOI: 10.1063/1.1416186

[64] NASA. NASA image gallery and NASA Glenn image galery. Last accessed: 2018-08-16. Retrieved from URLs:
https://www1.grc.nasa.gov/aeronautics/icing/airframe-icing/ (
https://www1.grc.nasa.gov/aeronautics/icing/#gallery.

[65] NATERER, G. F., "Eulerian three-phase formulation with coupled droplet flow and multimode heat transfer", *Numerical Heat Transfer*, vol. 43, no. 4, pp. 331–352, 2003. DOI: 10.1080/713836223

[66] NATIONAL TRANSPORTATION SAFETY BOARD (NTSB), "Aircraft accident report : In-flight icing encounter and loss of control simmons airlines, d.b.a. american eagle flight 4184 avions de transport regional (atr) model 72-212, n401am roselawn, indiana october 31, 1994", Aircraft Accident Report NTSB/AAR-96/01, Tech. Rep., Jul.. 1996.

[67] NORDE, E., "Eulerian method for ice crystal icing in turbofan engines", Ph. D. Thesis, University of Twente, Enschede, Jun. 2017.

[68] OSHER, S. and FEDKIW, R., *Level Set Methods and Dynamic Implicit Surfaces*, ANTMAN, S. S., MARSDEN, J. E., and SIROVICH, L., Eds. Springer, 2003.

[69] OSHER, S. and SETHIAN, J. A., "Fronts propagating with curvature-dependent speed: Algorithms based on hamilton-jacobi formulations", *Journal of Computational Physics*, vol. 79, no. 1, pp. 12–49, nov 1988. DOI: 10.1016/0021-9991(88)90002-2

[70] PENA, D., AL-KEBSI, A., HOARAU, Y., and LAURENDEAU, E., "Icing modeling on chimera grids using level-set approach", in *51st 3AF International Conference on Applied Aerodynamics*. 3AF, Apr.. 2016.

[71] PENA, D., HOARAU, Y., and LAURENDEAU, E., "Development of a three-dimensional icing simulation code in the nsmb flow solver", *International Journal of Engineering Systems Modelling and Simulation*, vol. 8, no. 2, pp. 86–98, 2016. DOI: 10.1504/IJESMS.2016.075544

[72] PENA, D., "Développement d'un code de givrage tridimensionnel avec méthode level-set", Ph. D. Thesis, Polytechnique Montreal, Montreal, May 2016.

[73] PIEGL, L. AND TILLER, W., *The NURBS Book*, series Monographs in Visual Communication. Springer Berlin Heidelberg, 2012.

[74] PLANTE, F., "Simulation de phénomènes instationnaires périodiques avec une méthode RANS dans le domaine fréquentiel", Master Thesis, Polytechnique Montréal, Apr.. 2017.

[75] POTAPCZUK, M. G. and REINMANN, J. J., "Icing simulation: A survey of computer models and experimental facilities", NASA Lewis Research Center, Tech. Rep., May 1991.

[76] POTAPCZUK, M., "Aircraft icing research at nasa glenn research center", *Journal of Aerospace Engineering*, vol. 26, no. 2, pp. 260–276, 2013. DOI: 10.1061/(ASCE)AS.1943-5525.0000322

[77] POTAPCZUK, M. and P. BROEREN, A., "An integrated approach to swept wing icing simulation", in *7th European Conference for Aeronautics and Space Sciences*. EUCASS Paper 2017-687, Jul.. 2017. DOI: 10.13009/EUCASS2017-687

[78] PUEYO, A., "Efficient 3D artificial ice shapes simulations with 2D ice accretion codes using a 3-level correction", in *SAE 2013 AeroTech Congress and Exhibition*, vol. 7. SAE Technical Paper 2013-01-2136, Sep. 2013. DOI: 10.4271/2013-01-2136

[79] PUEYO, A., CHOCRON, D., and KAFYEKE, F., "Improvements to the ice accretion code CANICE", in *8th CASI Aerodynamics Symposium*. CASI, May 2001.

[80] RADENAC, E., KONTOGIANNIS, A., BAYEUX, C., and VILLEDIEU, P., "An extended rough-wall model for an integral boundary layer model intended for ice accretion calculations", in *2018 Atmospheric and Space Environments Conference*. AIAA Paper 2018-2858, Jun. 2018. DOI: 10.2514/6.2018-2858

[81] RAUSCHENBERGER, P., CRISCIONE, A., EISENSCHMIDT, K., KINTEA, D., JAKIRLIĆ, S., Ž. TUKOVIĆ, ROISMAN, I., WEIGAND, B., and TROPEA, C., "Comparative assessment of volume-of-fluid and level-set methods by relevance to dendritic ice growth in supercooled water", *Computers & Fluids*, vol. 79, pp. 44 – 52, 2013. DOI: 10.1016/j.compfluid.2013.03.010

[82] RENDALL, T. and ALLEN, C., "Finite-volume droplet trajectories for icing simulation", *International Journal of Multiphase Flow*, vol. 58, pp. 185–194, jan 2014. DOI: 10.1016/j.ijmultiphaseflow.2013.08.007

[83] RTO/NATO, "Ice accretion simulation evaluation test", North Atlantic Treaty Organization, Tech. Rep., Nov. 2001.

[84] RUFF, G. A. and BERKOWITZ, B. M., "Users manual for the nasa lewis ice accretion code (LEWICE)", NASA, Tech. Rep., May 1990.

[85] SAAD, Y., *Iterative Methods for Sparse Linear Systems*, 2nd ed. Society for Industrial and Applied Mathematics, 2003.

[86] SADEK, H., "Aerodynamic design considerations fo flight in icing conditions", in *SAE 2007 Aircraft & Engine Icing International Conference & Exhibition*, Sep. 2007.

[87] SERKAN, O., ERHAN, T., and MURAT, C., "Parallel computing applied to three-dimensional droplet trajectory simulation in lagrangian approach", in *SAE 2011 International Conference on Aircraft and Engine Icing and Ground Deicing*. SAE Technical Paper 2011-38-0106, Jun. 2011. DOI: 10.4271/2011-38-0106

[88] SHIN, J., BERKOWITZ, B., CHEN, H., and CEBECI, T., "Prediction of ice shapes and their effect on airfoil performance", in *29th Aerospace Sciences Meeting*. American Institute of Aeronautics and Astronautics, Jan. 1991. DOI: 10.2514/6.1991-264

[89] SON, C., OH, S., and YEE, K., "Quantitative analysis of a two-dimensional ice accretion on airfoils", *Journal of Mechanical Science and Technology*, vol. 26, no. 4, pp. 1059–1071, 2012. DOI: 10.1007/s12206-012-0223-z

[90] SPALART, P. R. and ALLMARAS, S. R., "A one-equation turbulence model for aerodynamic flows", in *30th Aerospace Sciences Meeting and Exhibit*. AIAA Paper 1992-439, Jan. 1992. DOI: 10.2514/6.1992-439

[91] SWANSON, R. C., RADESPIEL, R., and TURKEL, E., "On some numerical dissipation schemes", *Journal of Computational Physics*, vol. 147, pp. 518–544, 1998. DOI: 10.1006/jcph.1998.6100

[92] SZILDER, K. and LOZOWSKI, E., "Progress towards a 3D numerical simulation of ice accretion on a swept wing using the morphogenetic approach", in *SAE 2015 Internation*

Conference on Icing of Aircraft, Engines and Structures. SAE Technical Paper 2015-01-2162, Jun. 2015. DOI: 10.4271/2015-01-2162

[93] SZILDER, K. and LOZOWSKI, E., "Three-dimensional numerical simulation of ice accretion on using a discrete morphogenetic approach", in *9th AIAA Atmospheric and Space Environments Conference.* AIAA Paper 2017-3418, Jun. 2017. DOI: 10.2514/6.2017-3418

[94] SZILDER, K. and YUAN, W., "The influence of ice accretion on the aerodynamic performance of a UAS airfoil", in *53rd AIAA Aerospace Sciences Meeting.* AIAA Paper 2015-536, Jan. 2015. DOI: 10.2514/6.2015-0536

[95] SZILDER, K. and LOZOWSKI, E. P., "Novel two-dimensional modeling approach for aircraft icing", *Journal of Aircraft*, vol. 41, no. 4, pp. 854–861, Jul.. 2004. DOI: 10.2514/1.470

[96] THOMPSON, D., TONG, C., ARNOLDUS, Q. *et al.*, "Discrete surface evolution and mesh deformation for aircraft icing applications", in *5th AIAA Atmospheric and Space Environments Conference.* AIAA Paper 2013-2544, Jun. 2013. DOI: 10.2514/6.2013-2544

[97] THOMPSON, D. S. and SONI, D. K., "ICEG2D: A software package for ice accretion prediction", in *41st AIAA Aerospace Sciences Meeting and Exhibit.* AIAA Paper 2003-1070, Jan. 2003. DOI: 10.2514/6.2003-1070

[98] TONG, X., THOMPSON, D., ARNOLDUS, Q., COLLINS, E., and LUKE, E., "Three-dimensional surface evolution and mesh deformation for aircraft icing applications", *Journal of Aircraft*, vol. 54, no. 3, pp. 1047–1063, 2017. DOI: 10.2514/1.C033949

[99] TRONTIN, P. and VILLEDIEU, P., "Revisited model for supercooled large droplet impact onto a solid surface", *Journal of Aircraft*, vol. 54, no. 3, pp. 1189–1204, 2017. DOI: 10.2514/1.C034092

[100] TRONTIN, P., BLANCHARD, G., KONTOGIANNIS, A., and VILLEDIEU, P., "Description and assessment of the new ONERA 2D icing suite IGLOO2D", in *9th AIAA Atmospheric and Space Environments Conference.* AIAA Paper 2017-3417, Jun. 2017. DOI: 10.2514/6.2017-3417

[101] VAHAB, M., PEI, C., HUSSAINI, M. Y., SUSSMAN, M., and LIAN, Y., "An adaptive coupled level set and moment-of-fluid method for simulating droplet impact and solidification on solid surfaces with application to aircraft icing", in *54th AIAA Aerospace Sciences Meeting*. AIAA Paper 2016-1340, Jan. 2016. DOI: 10.2514/6.2016-1340

[102] VASSBERG, J. C. and JAMESON, A., "In pursuit of grid convergence for euler and navier-stokes equations", *Journal of Aircraft*, vol. 47, no. 4, pp. 1152–1166, 2010. DOI: 10.2514/1.46737

[103] VICKERMAN, M., CHOO, Y., SCHILLING, H., BAEZ, M., DRAUN, B., and COTTON, B., "Toward an efficient icing CFD process using an interactive software toolkit - SmaggIce 2D", in *40th AIAA Aerospace Sciences Meeting and Exhibit*. AIAA Paper 2002-380, Jan. 2002. DOI: 10.2514/6.2002-380

[104] VILLEDIEU, P., TRONTIN, P., GUFFOND, D., and BOBO, D., "SLD lagrangian modeling and capability assessment in the frame of ONERA 3D icing suite", in *4th AIAA Atmospheric and Space Environments Conference*. AIAA Paper 2012-3132, Jun. 2012. DOI: 10.2514/6.2012-3132

[105] WRIGHT, W., "Further refinement of the LEWICE SLD model", in *44th AIAA Aerospace Sciences Meeting*. AIAA Paper 2006-464, Jan. 2006. DOI: 10.2514/6.2006-464

[106] WRIGHT, W. B., "Users manual for the improved nasa lewis ice accretion code LEWICE 1.6", NASA, Tech. Rep., May 1995.

[107] WRIGHT, W. B. and PORTER, C. E., "A revised validation process for ice accretion codes", in *9th AIAA Atmospheric and Space Environments Conference*. AIAA Paper 2017-3415, Jun. 2017. DOI: 10.2514/6.2017-3415

[108] WRIGHT, W. B., GENT, R. W., and GUFFOND, D., "DRA/NASA/ONERA collaboration on icing research: Part II–prediction of airfoil ice accretion", NASA Lewis Research Center, Tech. Rep., May 1997.

[109] WRIGHT, W. B., "Validation methods and results for a two-dimensional ice accretion code", *Journal of Aircraft*, vol. 36, no. 5, pp. 827–835, 1999. DOI: 10.2514/2.2516

[110] WRIGHT, W. B. and RUTKOWSKI, A., "Validation results for LEWICE 2.0", NASA/CR-1999-208690, Tech. Rep., Jan. 1999.

[111] XIA, H. and TUCKER, P. G., "Finite volume distance field and its application to medial axis transforms", *International Journal for Numerical Methods in Engineering*, pp. n/a–n/a, 2009. DOI: 10.1002/nme.2762

[112] ZHU, C., FU, B., SUN, Z., and ZHU, C., "3D ice accretion simulation for complex configuration basing on improved messinger model", in *International Journal of Modern Physics: Conference Series*, vol. 19. World Scientific, Jun. 2012, pp. 341–350. DOI: 10.1142/S2010194512008938

APPENDIX A HYPERBOLIC MESH GENERATION

This appendix contains the numerical developments of the 2D hyperbolic mesh generation algorithm which is also used as a basis for the hyperbolic ice accretion solver. The smoothing in the tangential and normal directions is also briefly presented. The development of the equations in 3D is very similar and can be found in the literature [25, 24].

Governing equations

The hyperbolic mesh generation is obtained by solving the following constraint equations (Eq. (A.1)) [51] to generate new layers of grid orthogonal to the previous layers with cells respecting as much as possible the contrait volume ΔV :

$$\mathbf{r}_\xi \cdot \mathbf{r}_\eta = 0 \tag{A.1a}$$

$$\hat{\mathbf{n}} \cdot (\mathbf{r}_\xi \times \mathbf{r}_\eta) = \Delta V \tag{A.1b}$$

By linearizing the constraint equations (Eq. (A.1)), a system of equations is obtained:

$$A_0 \mathbf{r}_\xi + B_0 \mathbf{r}_\eta = \mathbf{f} \tag{A.2}$$

where

$$A_0 = \begin{pmatrix} x_\eta & y_\eta \\ y_\eta & -x_\eta \end{pmatrix} \tag{A.3a}$$

$$B_0 = \begin{pmatrix} x_\xi & y_\xi \\ y_\xi & -x_\xi \end{pmatrix} \tag{A.3b}$$

$$\mathbf{f} = \begin{pmatrix} 0 \\ \Delta V \end{pmatrix} \tag{A.3c}$$

This system is solved by marching in the η direction which acts as a time variable in convective equations according to Chan and Steger [25]. The system is then discretized using a non-iterative implicit finite difference scheme which is unconditionally stable and allows to choose the marching step in η depending on grid quality considerations only. Discretizing the tangential derivative (ξ-direction) with a central spatial difference and the normal derivative

(η-direction) with a two points backward implicit difference, Eq. (A.2) is rewritten as:

$$A_l \delta_\xi \left(\mathbf{r}_{l+1} - \mathbf{r}_l\right) + B_l \nabla_\eta \mathbf{r}_{l+1} = \mathbf{f}_{l+1} \tag{A.4}$$

where the central difference is expressed as $\delta_\xi \mathbf{r}_i = \frac{1}{2}(\mathbf{r}_{i+1} - \mathbf{r}_i)$ and the backward difference is expressed as $\nabla_\eta \mathbf{r}_{l+1} = \mathbf{r}_{l+1} - \mathbf{r}_l$. Note that i and l are the grid indices in the ξ- and η-directions respectively. Eq. (A.4) is then multiplied by B_l^{-1} and approximately factored. Numerical dissipation in the ξ-direction is also added explicitly and implicitly to compensate for the central differencing. The complete system of equation with additional smoothing and implicitness is thus written in delta form as:

$$\left[I + (1+\theta) B_l^{-1} A_l \delta_\xi - \varepsilon_i \left(\Delta\nabla\right)_\xi\right] \Delta\mathbf{r}_l = B_l^{-1} f_{l+1} - \varepsilon_e \left(\Delta\nabla\right)_\xi \mathbf{r}_l \tag{A.5}$$

where θ increases implicitness and prevents grid lines crossings in concave regions, ε_e and ε_i are the explicit and implicit smoothing coefficients respectively which are detailed in the next section. The second order differencing operator is expressed as $\left(\Delta\nabla\right)_\xi = \mathbf{r}_{i+1} - 2\mathbf{r}_i + \mathbf{r}_{i-1}$.

Derivatives

The derivatives $\mathbf{r}_\xi$ and $\mathbf{r}_\eta$ are computed with two different methods. First, the derivative components in the ξ direction are approximated with central differencing such as mentioned earlier. As for the η direction, the derivative components are obtained by directly solving the constraint equations (Eq. (A.1)) without linearization :

$$x_\eta = -\frac{y_\xi \Delta V}{J} \tag{A.6a}$$

$$y_\eta = \frac{x_\xi \Delta V}{J} \tag{A.6b}$$

where $J = x_\xi^2 + y_\xi^2$ is the squared Jacobian of matrix B. This definition of the η-derivative can lead to incorrect definition in corners made of unequally spaced points, in which case a correction is needed. The reader is referred to Chan and Steger [25] for further details on the subject.

Smoothing

The most crucial point in the hyperbolic mesh generation is the different grid smoothing mechanisms that are added to stabilize the system. In the diffusion term from the left-hand side of Eq. (A.5), the coefficient ε_e depends on five quantities which handle the smoothing in different conditions.

$$\varepsilon_e = \varepsilon_c N_\xi S_l d_\xi a_\xi \tag{A.7}$$

From these quantities, only ε_c is adjustable by the user while the others are computed automatically by the scheme. The effects of the different quantities are described below, but the reader is referred to Chan and Steger [25] and Chan [24] for details on the actual equations.

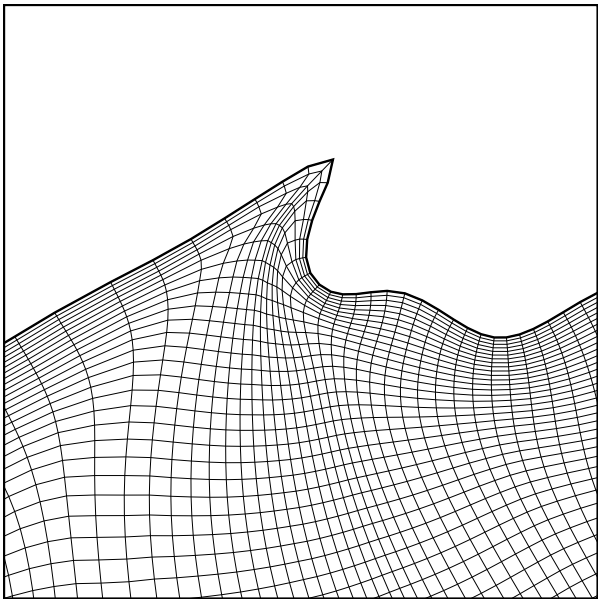
1. As mentioned, ε_c is defined by the user with $O(1)$. Typically, a value of 0.5 is enough on simple geometries .
2. A scaling of the metrics regarding the local mesh spacing is provided by N_ξ.
3. S_l acts as a scaling function controlling the level of smoothing depending on the distance to the solid body from zero near the body to one at the outer boundary. This allows to preserve orthogonality near the body.
4. The grid convergence sensor d_ξ is used to increase locally the smoothing in concave regions to avoid the convergence of grid lines. This mechanism depends on the average distance between grid points at levels $(l-1)$ and l and limiters are used to prevent too low values in convex regions.
5. The grid angle function a_ξ is used to locally increase the smoothing at severe concave corners, as opposed to entire concave regions with the parameter d_ξ.

Other correction methods are required to ensure a robust mesh generation, such as the handling of convex corners where the actual system is replaced in the global matrix by an average of neighbours. The boundary conditions are also important to consider, as well as the specification of cell sizes. However, these subjects go beyond the scope of this work and the interested reader is referred to the already mentioned literature for further details.

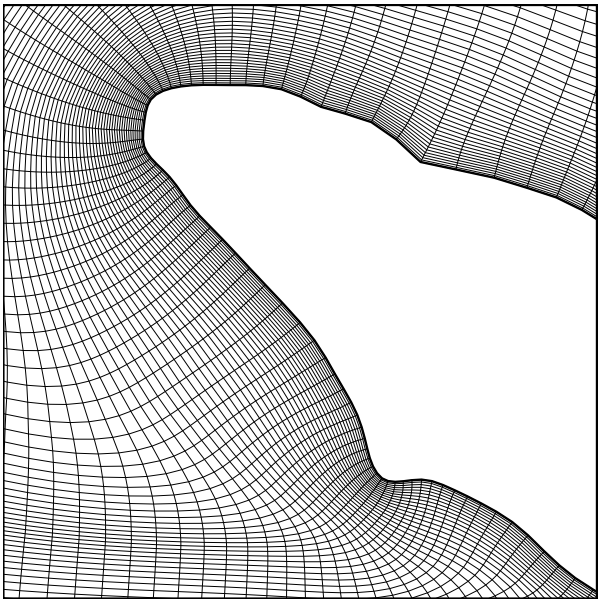
Results

Figure A.1(c) shows a mesh generated with the hyperbolic mesh generation method on an experimental ice shape [100]. Although the mesh metrics are not studied quantitatively, it can be observed that the mesh is of high quality even though the shape is highly deformed

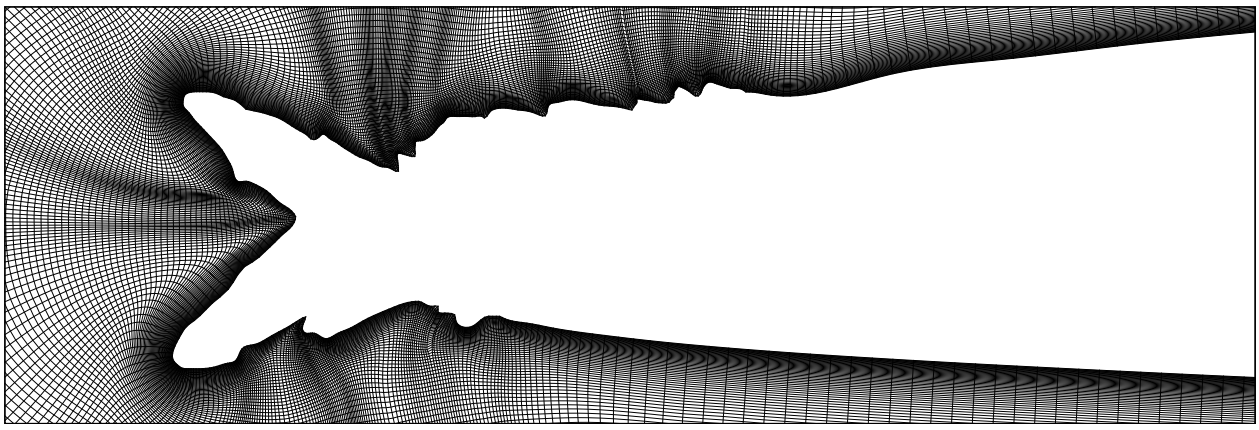
and contains small details. Grid lines convergence can be seen, but the different smoothing mechanisms of the method avoid strong grid shocks from being generated. In particular, Figure A.1(a) shows how sharp concavities are well handled and Figure A.1(b) shows how the grid in convex regions is generated orthogonally to the geometry. Figures A.2(a) to A.2(d) present the orthogonality, skewness, aspect ratio and J-stretch ratio grid quality metrics around the ice shape in the form of contour plots, with the objective being 0 for the skewness and 1 for the other metrics. The small number of cells coloured in light green in the figures testifies that the grid is generated with high quality, as the bad quality cells are coloured in dark green.



(a) Hyperbolic grid in concave region

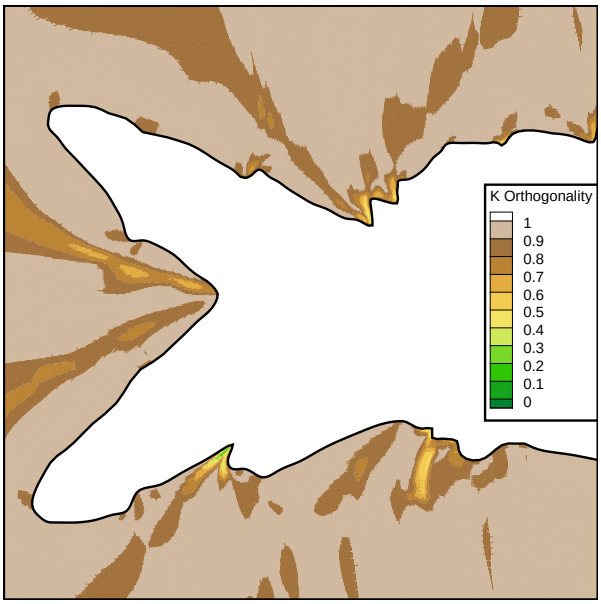


(b) Hyperbolic grid in convex region

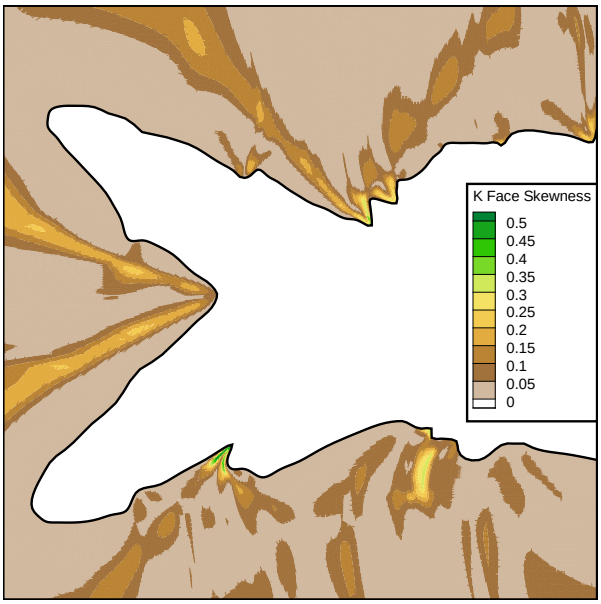


(c) Hyperbolic grid on experimental ice (GLC305)

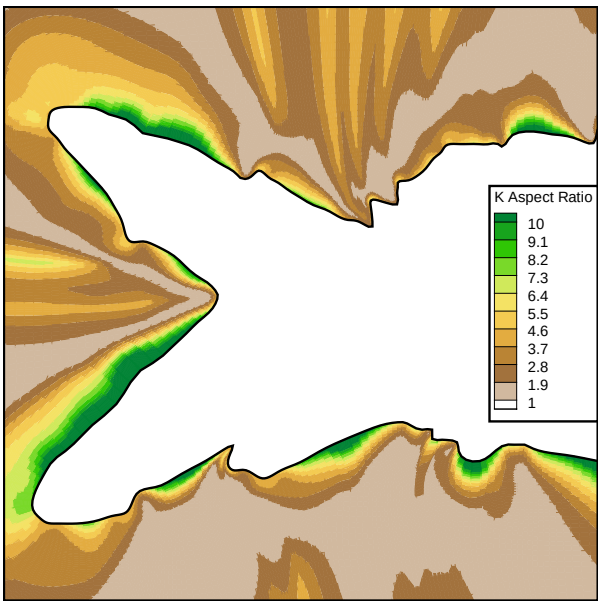
Figure A.1 Hyperbolic mesh generation example



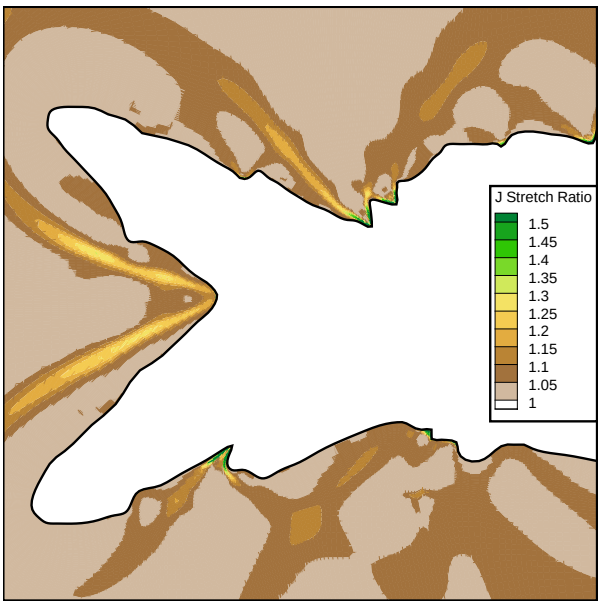
(a) Hyperbolic grid orthogonality criterion



(b) Hyperbolic grid skewness criterion



(c) Hyperbolic grid aspect ratio criterion



(d) Hyperbolic grid J-stretch ratio criterion

Figure A.2 Hyperbolic mesh quality metrics

APPENDIX B COMPARATIVE STUDY DATA

This appendix contains the quantitative metrics data used to compute the error values in Chapter 6 for the two icing cases studied, as well as the metrics obtained for the experiment.

Experiments

Table B.1 contains the different quantitative metrics computed for the experimental data of glaze ice case 401 and rime ice case 405.

Table B.1 Quantitative metrics : Experiments

Case		**Glaze case 401**	**Rime case 405**
Ice area	[m^2]	4.89E-04	5.79E-04
Upper horn length	[m]	1.84E-02	1.56E-02
Upper horn angle	[°]	29.21	-19.85
Lower horn length	[m]	9.94E-03	N/A
Lower horn angle	[°]	-61.99	N/A
Ice coverage	[m]	1.20E-01	1.01E-01

Glaze ice case 401

The quantitative metrics of the glaze ice case 401 obtained for the algebraic, level-set and hyperbolic methods are shown in Table B.2, Table B.3 and Table B.4 respectively.

Table B.2 Glaze case 401 quantitative metrics : Algebraic method

Number of layers		1	5	10	20
Numerical area	[m^2]	5.43e−4	4.52e−4	4.90e−4	4.38e−4
Predicted area	[m^2]	5.38e−4	4.70e−4	4.90e−4	4.37e−4
Upper horn length	[m]	1.63e−2	1.61e−2	2.05e−2	2.26e−2
Upper horn angle	[°]	30.57	12.33	6.78	2.84
Lower horn length	[m]	9.28e−3	1.02e−2	1.06e−2	1.04e−2
Lower horn angle	[°]	−73.69	−60.85	−57.63	−58.27
Ice coverage	[m]	8.50e−2	8.78e−2	8.73e−2	8.99e−2

Table B.3 Glaze case 401 quantitative metrics : Tracked level-set method

Number of layers		1	5	10	20
Numerical area	[m^2]	5.38e−4	5.53e−4	5.22e−4	4.26e−4
Predicted area	[m^2]	5.38e−4	5.50e−4	5.23e−4	4.36e−4
Upper horn length	[m]	1.27e−2	2.08e−2	2.52e−2	2.31e−2
Upper horn angle	[°]	34.35	30.8	10.82	−0.65
Lower horn length	[m]	6.92e−3	9.79e−3	1.14e−2	1.26e−2
Lower horn angle	[°]	−65.93	−66.07	−70.41	−67.88
Ice coverage	[m]	8.74e−2	8.98e−2	8.78e−2	8.89e−2

Table B.4 Glaze case 401 quantitative metrics : Hyperbolic method

Number of layers		1	5	10	20
Numerical area	[m^2]	5.27e−4	4.80e−4	4.44e−4	3.99e−4
Predicted area	[m^2]	5.38e−4	4.80e−4	4.43e−4	3.98e−4
Upper horn length	[m]	1.03e−2	1.85e−2	2.01e−2	2.01e−2
Upper horn angle	[°]	22.7	10.03	3.41	14.56
Lower horn length	[m]	8.85e−3	1.15e−2	1.18e−2	1.31e−2
Lower horn angle	[°]	−63.53	−64.76	−61.05	−58.84
Ice coverage	[m]	8.82e−2	8.93e−2	9.00e−2	9.26e−2

Rime ice case 405

The quantitative metrics of the rime ice case 405 obtained for the algebraic, level-set and hyperbolic methods are shown in Table B.5, Table B.6 and Table B.7 respectively.

Table B.5 Rime case 405 quantitative metrics : Algebraic method

Number of layers		1	5	10	20
Numerical area	[m^2]	5.79e−4	5.18e−4	5.37e−4	5.29e−4
Predicted area	[m^2]	5.73e−4	5.46e−4	5.37e−4	5.31e−4
Main horn length	[m]	1.22e−2	1.46e−2	1.76e−2	1.87e−2
Main horn angle	[°]	−32.64	−17.45	−9.69	−14.32
Ice coverage	[m]	8.93e−2	1.10e−1	1.10e−1	1.11e−1

Table B.6 Rime case 405 quantitative metrics : Tracked level-set method

Number of layers		1	5	10	20
Numerical area	[m^2]	5.73e−4	5.50e−4	5.41e−4	5.30e−4
Predicted area	[m^2]	5.73e−4	5.51e−4	5.42e−4	5.33e−4
Main horn length	[m]	1.12e−2	1.61e−2	1.75e−2	1.84e−2
Main horn angle	[°]	−36.11	−19.27	−12.66	−13.7
Ice coverage	[m]	9.25e−2	1.11e−1	1.09e−1	1.12e−1

Table B.7 Rime case 405 quantitative metrics : Hyperbolic method

Number of layers		1	5	10	20
Numerical area	[m^2]	5.72e−4	5.45e−4	5.33e−4	5.28e−4
Predicted area	[m^2]	5.73e−4	5.46e−4	5.34e−4	5.31e−4
Main horn length	[m]	1.13e−2	1.63e−2	1.78e−2	1.88e−2
Main horn angle	[°]	−38.68	−15.19	−11.79	−13.85
Ice coverage	[m]	9.25e−2	1.07e−1	1.03e−1	1.06e−1